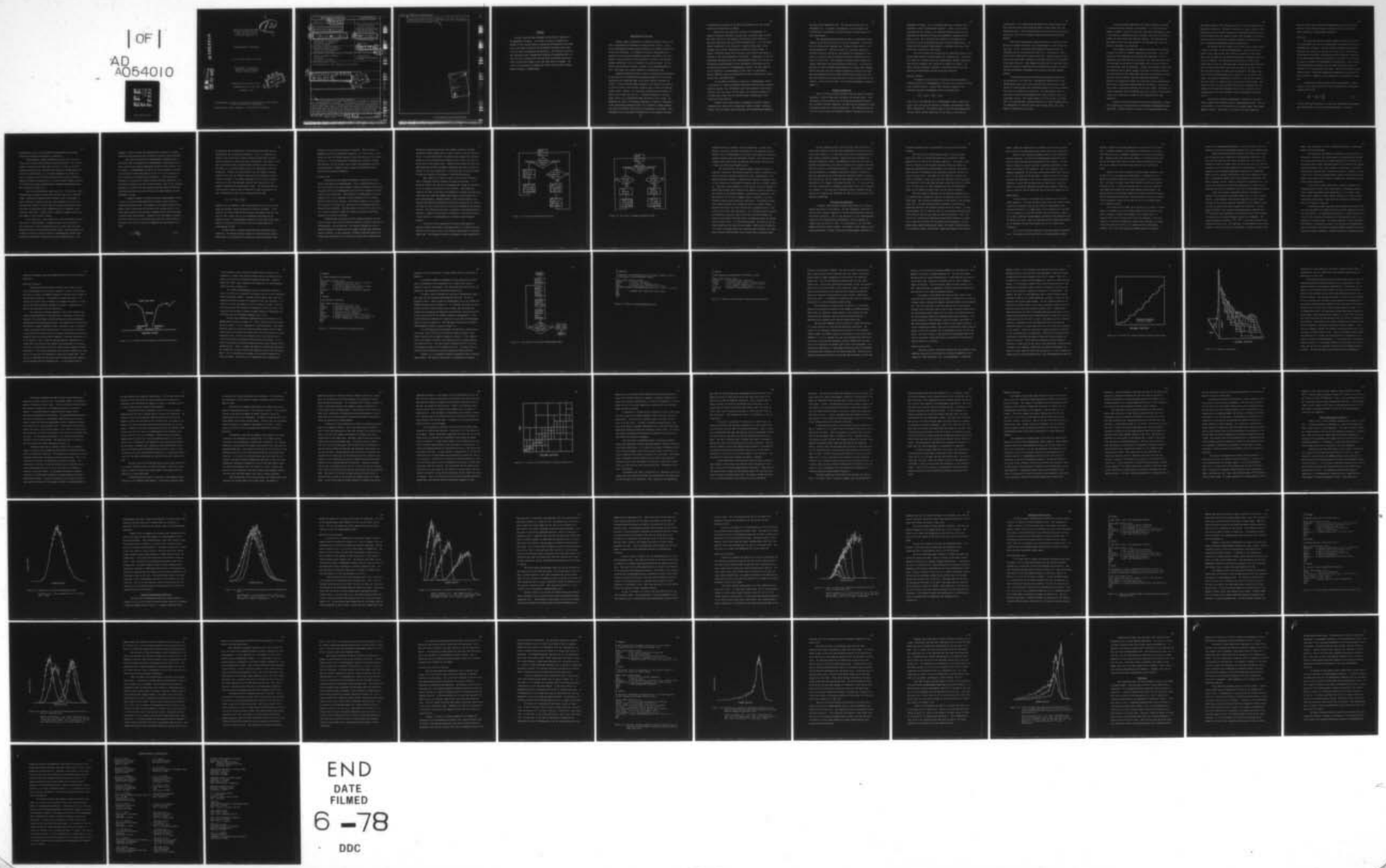
AD-A054 010 ARIZONA UNIV TUCSON DEPT OF CHEMISTRY F/G 9/2
CHROMATOGRAPHY SIMULATION.(U)
FEB 77 J B PHILLIPS, M F BURKE N00014-75-C-0512
UNCLASSIFIED TR-11 NL

| OF |
AD
A054010



END
DATE
FILMED
6 -78
DDC

12

OFFICE OF NAVAL RESEARCH
Contract N00014-75-C-0512
Task No. NR 051-518
Technical Report No. 11

CHROMATOGRAPHY SIMULATION

J. B. Phillips and M. F. Burke

Department of Chemistry
University of Arizona
Tucson, Arizona 85721

Submitted for Publication in

Communications of the ACM

February 1977

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER Number 11 ✓ (14) TR-11 2. GOVT ACCESSION NO.	3. RECIPIENT'S CATALOG NUMBER
4. TITLE (and Subtitle) (6) CHROMATOGRAPHY SIMULATION.	5. TYPE OF REPORT & PERIOD COVERED Technical Report 2-1-74 10-31-77 6. PERFORMING ORG. REPORT NUMBER N00014-75-C-0512
7. AUTHOR(s) (10) J. B. Phillips [redacted] M. F. Burke	8. CONTRACT OR GRANT NUMBER(s) (15) N00014-75-C-0512 ✓
9. PERFORMING ORGANIZATION NAME AND ADDRESS Department of Chemistry ✓ University of Arizona Tucson, AZ 85721	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS NR051-518
11. CONTROLLING OFFICE NAME AND ADDRESS Material Sciences Division Office of Naval Research Arlington, VA 22217	12. REPORT DATE (11) Feb 77 (12) 13. NUMBER OF PAGES 79 83 p.
14. MONITORING AGENCY NAME & ADDRESS (if different from Controlling Office) Office of Naval Research (Code 472) Arlington, VA 22217	15. SECURITY CLASS. (of this report) Unclassified 15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Distribution of this document is unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

(9) Technical rept. 1 Feb 74-31 Oct 77,

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

A unique system has been developed for the digital simulation of chromatographic processes. This system is based on a probabalistic approach to the discrete events of adsorption and desorption rather than using a continuous solution of the differential equations used to describe the rates of adsorption and desorption. The simulation system has been developed using a threaded code technique of programming which allows the user to interact with the high speed, microcoded internal portions of the system through → next page

033 860

a very high level specific language. The utility of the system for studying both linear and non-linear chromatographic processes is demonstrated.

ABSTRACT

A unique system has been developed for the digital simulation of chromatographic processes. This system is based on a probabalistic approach to the discrete events of adsorption and desorption rather than using a continuous solution of the differential equations used to describe the rates of adsorption and desorption. The simulation system has been developed using a threaded code technique of programming which allows the user to interact with the high speed, microcoded internal portions of the system through a very high level specific language. The utility of the system for studying both linear and non-linear chromatographic processes is demonstrated.

CHROMATOGRAPHY SIMULATION

Digital computer simulation is a powerful technique useful as an aid in understanding the behavior of complex systems (50-53). A gas-solid chromatography column certainly qualifies as a complex system which is not well understood. Its behavior is basically determined by the chemistry of the gas-solid adsorption processes occurring in the column. However, these processes are only indirectly connected to the chromatographic experimental results, retention time, and peak shape. The connection must be made by a theory, or model, of the chromatographic process. Such models are usually expressed in mathematical terms.

Computer simulation provides a means of connecting the mathematics of the model to the experimentally measurable properties of the system, showing how the assumptions made in the model logically determine its results. If the cause and effect sequences in the model are the same as in the real system, then the model should be able to mimic the behavior of the system. However, it is not always obvious what the behavior of a model, even a simple one, will be. A computer simulation is the most direct way to test a mathematical model and find out how it works. By comparing the results of simulation experiments on models to experiments on the corresponding chemical system, it is possible to derive information about the chemical system and its relationship to the models. This information may be qualitative determinations of the adequacy of models or quantitative estimates of the values of parameters that the chemical system and its model have in common.

Most of the basic processes involved in chromatography, for example, adsorption, diffusion, and gas flow, are fairly simple to model. But, when they are all put together, the resulting system behavior is much more complicated than any of the individual parts. It is the simultaneous interaction of all the parts or processes which makes chromatography such a useful technique and a difficult system to model. A number of different models may be involved, each of which contributes in some way to the overall behavior of the system. A simulation is a way of performing experiments upon these chromatographic models with the goal of developing an understanding of their simultaneous interactions and the corresponding processes in chromatographic systems. The most important use of simulation in chromatography at this time is to distinguish between different models and combinations of models rather than to derive results for a particular model.

As the number of processes involved in a chromatographic system is increased, the number of possible interactions between models increases much more rapidly, soon reaching the point where analytical mathematical solutions can no be found. At this point, the electronic computer with its high speed becomes the only feasible means of performing a fully integrated system analysis.

Because of the large amounts of computation involved, computer simulations are usually run on large-scale computer systems. Recently, however, there has been a trend toward the use of dedicated minicomputers for large-scale computations (54). The two major reasons for this development are: 1) economic, minicomputers are much cheaper; and 2) convenience, minicomputers are often already available because of other applications.

The computation described here was done on a medium-scale minicomputer system for both reasons of cost and convenience. One critical part of the particular computer used, a Hewlett-Packard 2100A, is its microprogrammability. Coding key parts of a simulation program in microcode typically reduced the time required for computation of these parts by a factor of ten and reduced the overall simulation time by a factor of two to five. This increase in execution speed allowed more complex simulations than would otherwise be practical. It also made it practical to code the models being simulated in a higher level language. Models expressed using this threaded programming technique are so much more understandable that the figures defining models simulated in this work are presented in the same form used to code and present them to the computer. More will be said about this threaded programming technique in Chapter 4.

Methods of Simulation

There are basically three different kinds of computer simulation techniques: analytic functions, continuous, and discrete event. These correspond to three different ways of formulating mathematical models. For an analytic function simulation, the model must be described in terms of equations which can be solved analytically. The simulation then consists of a set of values for dependent variables calculated from sets of independent variables. For a continuous simulation, the model must be described in terms of differential equations. Given a set of initial conditions for the variables, the simulation then moves the system in simulated time numerically solving the differential equations at each point in time. For a discrete event simulation, the model must be described in terms of mechanisms and even occurrence probabilities. The simulation then produces random numbers to determine what events occur and collects statistics on the results.

The kind of model best representing the important features of the system being simulated should determine the simulation type. The choice of a model for a complex system like chromatography depends largely upon what kind of information the simulation is supposed to produce. All three, analytic functions, continuous, and discrete event, have been applied to chromatographic systems in various situations.

Analytic Functions

An example of the utility of an analytic function simulation technique research in chromatography is the work done by Chesler and Cram (55) on moment analysis. Using a small laboratory computer, they generated simulated chromatograms with the functional form

$$y(x) = y_g(x) + y_t(x) + y_e(x)$$

where $y(x)$ is the amplitude of the chromatographic peak at time x and $y_g(x)$, $y_t(x)$, and $y_e(x)$ are Gaussian, triangular, and exponential functions, respectively. The Gaussian alone is used for the leading side of the peak, while various combinations of all three are used for the trailing side. It was empirically determined that simulated peaks with this functional form closely match the shape of some typical real chromatographic peaks. These simulated peaks were used to test the precision and accuracy with which the moments could be calculated using various methods.

Only three simulated peak shapes were used so it would have been possible to generate them from a real chromatographic system without too much work. In fact, real peaks were produced and compared with the simulated peaks to see how realistic they were. The reason for simulation in this example was not just to save time. Having an analytic function for the peaks made it possible to calculate exactly what the moments of the peaks should be. Then moments calculated for the same peaks using various techniques of sampling from the simulated data could be compared with the true moments to determine the accuracy of the data sampling methods.

A second application of simulation using analytic functions is the generation of test data (56). To adequately debug and test a new chromatographic data system, the system must be exercised with a large variety of chromatographic data containing examples of all the situations that might be encountered in actual use. The required data could be generated by digitizing chromatograms from a wide variety of real chromatographic systems. This would, however, be a lot of work. For each test, the chromatographic system would have to be tuned to produce all of the combinations of retention times, peak overlap, noise, etc., that might be encountered in real data.

In this particular application, an analytic function simulation is a much more direct solution to the problem. It is much faster and simpler to supply a series of values for a few empirical parameters such as the number of components and their retention times than it would be to produce the same kind of data from real mixtures. The simulation can also produce better test data because it can systematically cover all possible combinations of situations.

This example illustrates two important advantages of simulation in general and analytic function simulation in particular. First, the time scale of an experiment can be drastically compressed. Not all simulations run faster than the real system. But, even when they do not, the experiment set up may be so much simpler that it is still faster to do an experiment with the simulation. Second, the parameters controlling the simulation may be easily changed to precisely the values desired to test the behavior of the system under conditions not conveniently attainable with the real system. With both speed and convenience, it is possible to interactively explore the system under widely varying conditions to get a qualitative understanding of its behavior. This can be very useful in designing experiments for the real system. Eventually, of course, experiments must be performed using real systems, but preliminary testing with simulations can make the real experiments much more productive.

Analytic function simulation is also widely proposed as a method for recognizing and assigning areas to overlapping chromatographic peaks (15,57-66). The basic idea is to find a set of simulated peaks which, when added together, will reproduce the shape of the real chromatogram as accurately as possible. The area of each peak can then be found by integrating the analytic function defining the corresponding simulated peak. If the simulated peaks are properly chosen, this procedure is potentially much more accurate than the more empirical methods of assigning areas such as dropping perpendiculars between adjacent peaks.

The problem with this method is that there are an infinite number of ways to fit peaks to any given chromatogram. Restrictions must be placed on the kinds of curve fits which are acceptable so that, hopefully, the one picked by the computer program will correspond closely to the real situation. The restrictions are generally of three types: minimizing the number of simulated peaks, eliminating illegal situations such as negative peak areas, and fixing the shape of the peaks. The first two restrictions are fairly easy to implement. The proper shape for a simulated chromatographic peak, however, is not so easily determined. The simplest approximation, Gaussian, is rarely close enough to the true shape to be properly used in this kind of curve fitting. Much effort has gone into devising functional forms for simulated peaks and methods of adjusting their parameters to give the truest fit to real chromatograms in the widest possible range of experimental conditions (67-71).

The analytic function chromatographic simulation is concerned with the empirically determined shape of chromatographic peaks. Thus, it can be useful as a tool for working with the output signals from chromatographic systems. Since there is no direct connection between the functional forms used in fitting chromatographic peaks and the column processes which produced the peaks, additional theory is required to obtain fundamental chromatographic parameters.

Continuous

The major application of digital computer continuous simulation in chromatography has been in the development and testing of theories of chromatographic processes. Two very important simplifying assumptions have been used in most such theories in the past (72). In the linear chromatography assumption, the partition ratio or fraction of molecules in the mobile phase divided by the fraction of molecules in the stationary phase is assumed to be independent of concentration. In the equilibrium chromatography assumption, the stationary and mobile phases are assumed to be at equilibrium at all times. There can be no mass transfer limited effects. Many other simplifying assumptions are also used but none of them is as important or has received as much attention as these two.

Assuming both linear and equilibrium chromatography, a simple equation can be derived from Fick's laws of diffusion for the chromatographic process (73):

$$\frac{\partial C}{\partial t} = - V \frac{\partial C}{\partial z} + D \frac{\partial^2 C}{\partial z^2} \qquad (3.1)$$

In this differential equation, C is the total concentration (stationary plus mobile phase) at a time t and column position z. The two coefficients, V and D, are the effective chromatographic peak drift velocity and diffusion coefficients, respectively.

Chromatographic systems described by equation (3.1) are relatively easy to solve using continuous simulation methods. The simulated column is divided up into small units of size Δz. These are like the theoretical plates of the well-known plate theory of chromatography (74) in that the concentration, C, is the same throughout the volume of a unit, Δz; however, they are really derived from the finite differences method of simulating equation (3.1). A value for HETP (Height Equivalent to a Theoretical Plate) calculated from a simulated chromatogram would not necessarily be equal to Δz.

Time is also divided into small units, Δt. At time zero, a given concentration of sample is placed into the first unit of the simulated column. During each simulation time increment, some of the sample in each column unit is moved to the next unit, depending on the flow rate and partition constant, and some sample is diffused from high concentration units to neighboring lower concentration units, depending on the diffusion coefficient. Finally, after a sufficient number of time units, the sample is eluted from the column.

This simulation is quite simple, but it does illustrate the basic principles of chromatography. Simulated mixtures can be separated by their differential rates of migration down the column, while the peaks gradually broaden into the expected Gaussian shape. Many refinements can be added to make more realistic models of chromatographic systems while keeping the same simple structure for the basic simulation (75). The computer is able to execute the simulation fast enough to be somewhat interactive and the results can be related to realistic input parameters.

The linear and equilibrium chromatography assumptions made in this model limit the simulation to chromatographic systems which are at least close to ideal and for applications where non-ideal behavior is of no interest. Chromatographic systems of the type described by equation (3.1) have been treated analytically (76,77) so it is not necessary to use a simulation to calculate their properties. If the equilibrium assumption is removed, the situation becomes much more complicated, but satisfactory solutions can still be found (78,79). But, if the linearity assumption is removed (72), the analytic solutions become so difficult and complicated that they are of little use in understanding real chromatographic systems.

A computer program to simulate nonlinear chromatographic systems is only a little more complicated than the simple linear continuous simulation described above. The equations to calculate how much of the sample to move from one unit to the next on the column are just a bit more complex because of the nonlinear isotherm used to relate stationary to mobile phase concentrations. Equation (3.2), the Langmuir isotherm, describes adsorption on a stationary phase with only a finite number of adsorption sites each of which can accommodate only one molecule at a time:

$$C_s = \frac{K_1C_m}{1 + K_2C_m} \qquad (3.2)$$

C_s and C_m are the concentrations in the stationary and mobile phases, respectively, and K_1 and K_2 are constants. If $K_2 = 0$, equation (3.2) reduces to the linear case in which a constant proportion of the molecules are adsorbed no matter the total concentration. This makes it very easy to calculate using a single multiply instruction what fraction of the sample is in the mobile phase ready to be transferred to the next column unit. However, for the nonlinear case, the Langmuir isotherm, equation (3.2), must be solved at each step of the simulation using an iterative procedure. The computer program to do so is not very complicated, but it does take much more time than the linear case so the Langmuir simulation runs significantly slower. The execution speed of this nonlinear simulation can be increased by replacing equation (3.2) with an approximation which can be calculated faster (80):

$$C_s = K_0 + K_1C_m + K_2C_m^{\ 2} \qquad (3.3)$$

Equation (3.3) is parabolic rather than hyperbolic and so can be solved using the quadratic formula instead of an iterative procedure. If the constants are chosen correctly and the total concentration does not vary over too wide a range, then it is an acceptable substitute for the Langmuir equation. Chromatography simulations of this type using various nonlinear isotherms have been widely used for testing theories of chromatography (81-86).

To most chemists, computer simulation means continuous system simulation. The analytic function variety of simulation is not complicated enough to be considered real simulation, while the discrete event variety is just not used very often in chemistry. When the model is expressed in terms of differential equations, as it very often is, then using the finite differences approach is the most obvious way to simulate and test it. But the mathematics of differential equations, although very powerful, are not the only way to model chromatographic processes. For some purposes, different kinds of models and simulations may be useful and so should be considered.

Discrete Event

The discrete event computer simulation technique has not previously been applied to chromatography. This is probably because it is not as well known as the other computer simulation techniques and because the usual theories of chromatography seem to fit more naturally into continuous system simulations. Most theories of chromatography start with equation (3.1) or its equivalent and then elaborate on it to include whatever effects are of interest. One notable exception is the random walk approach of Giddings (87), which starts with individual molecules making discrete movements. His basic model is idealized in such a way so as to lead to equations suitable for analytic function or continuous simulation as quickly as possible.

Chromatography itself has not been simulated using the discrete event technique, but some of the physical processes which are involved in chromatographic systems have been. For example, Nakagawa has written computer programs to simulate both the Langmuir and BET type adsorption processes (88,89). In these programs, a computer-simulated surface with 10,000 empty adsorption sites is placed in contact with a simulated gas.

During each simulation time unit, the computer generates a uniform distribution random number which is used to select one of the adsorption sites. A second random number is generated and compared with the probabilities for an adsorption and desorption event to make a decision on which type of event should occur. Simplified flow charts for these programs modelling the BET and Langmuir adsorption processes are given in Figures 3.1 and 3.2, respectively. A record is kept of the total number of molecules adsorbed at each point in time during simulation.

This simulation is useful in that it very clearly illustrates what is happening during a nonlinear non-equilibrium process. It is possible to derive the same results mathematically though the derivation is not as simple and intuitive as the simulation. When additional features are added to the model, the mathematics involved can become much worse while the simulation program only becomes a little bigger but is still understandable. Nakagawa (88,89), using similar programs, also simulated the adsorption-desorption process for surfaces with two different kinds of sites and with access to some sites hindered by the presence of pores. The simulation with pores is especially interesting because it shows how a complex structure which is difficult to accurately describe in most theories of chromatography can be modeled using a computer program.

The discrete event approach has been most often applied to queueing networks and similar system models where it is obvious that the behavior of the system is due to the nonlinear interaction of individual events (90). The analogous situation in chemistry is the interaction of

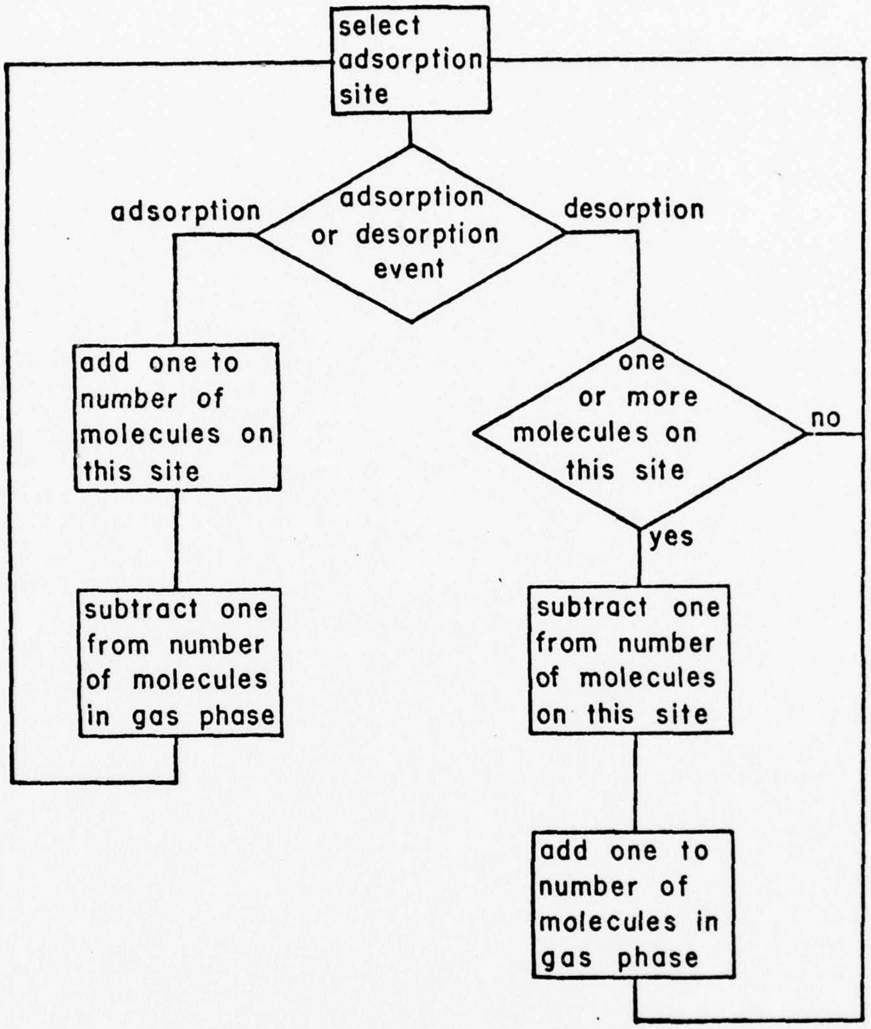

select adsorption site
adsorption
adsorption or desorption event
desorption
add one to number of molecules on this site
one or more molecules on this site
no
yes
subtract one from number of molecules in gas phase
subtract one from number of molecules on this site
add one to number of molecules in gas phase

Figure 3.1 Flow chart of BET adsorption model.

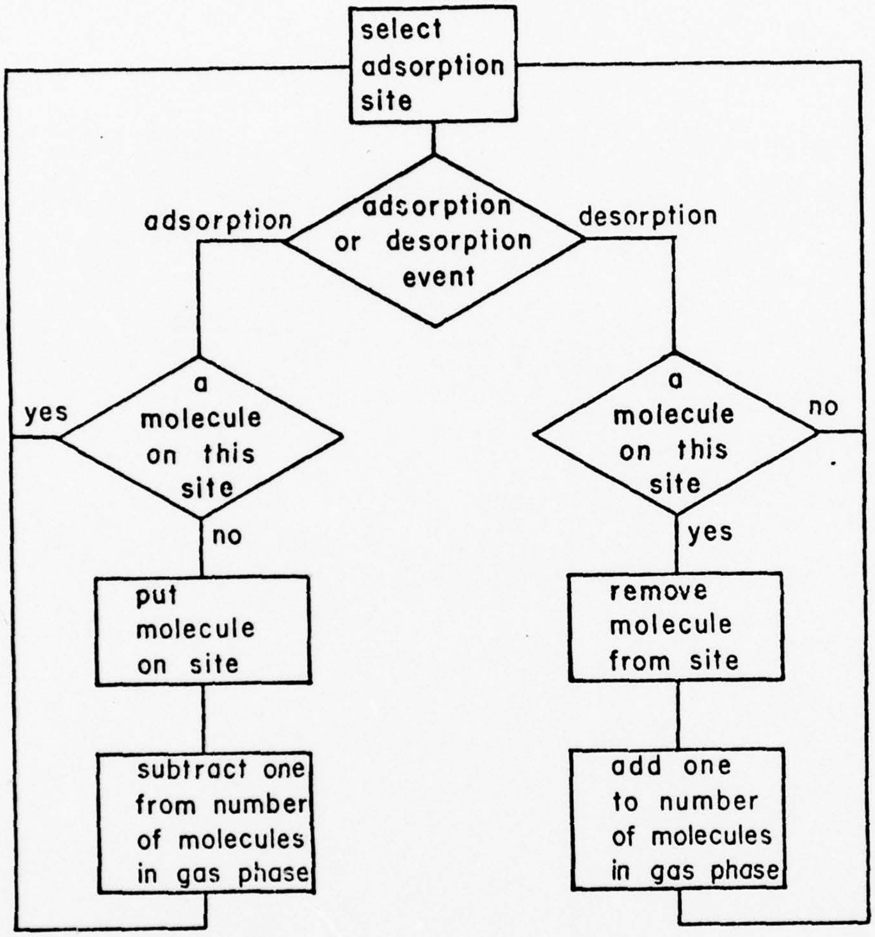

select adsorption site
adsorption
adsorption or desorption event
desorption
yes
a molecule on this site
no
a molecule on this site
no
yes
put molecule on site
remove molecule from site
subtract one from number of molecules in gas phase
add one to number of molecules in gas phase

Figure 3.2 Flow chart of Langmuir adsorption model.

individual molecules through a series of mechanisms. In many cases, chemists are able to calculate the behavior of such chemical processes from the laws of thermodynamics and discrete event models are confined to informal reasoning about the mechanisms involved. But, discrete event simulation has proved useful in modelling some chemical processes; for example, the combustion of gasoline (91).

The three varieties of digital computer simulation (analytic functions, continuous, and discrete event) are useful in different situations and require different techniques for their implementation. The analytic functions approach is generally the simplest to implement since it just involves the calculation of formulas with sets of numbers plugged in. Continuous system simulation is next because it has the problems of maintaining accuracy while simulating the passage of time in addition to the problems of calculating formulas. Discrete event simulation is the most difficult to fit into a computer since, in addition, it has the problems of maintaining statistical significance. The amount of detail in an underlying model for a simulation varies in a similar fashion. A discrete event model is rich in descriptions of mechanisms and is intuitively easy to think about and discuss. In a continuous model described by differential equations, much of the detail has been averaged out. Instead of describing individual events, the model describes the average result of many events. In a curve-fitting model, the detail of how the system progresses with time has been removed. Another way of looking at it is that a continuous model can be derived from a discrete event model and an analytic function model can be derived from a continuous model.

The most important reason a discrete event model was chosen to simulate the behavior of the gas-solid chromatographic system is because, of the three varieties, it is the most natural way of thinking about the basic chemical processes involved. Complex nonlinear adsorption and desorption mechanisms inside a chromatographic column are much easier to describe and understand in the form of discrete event computer algorithms than they would be as differential equations. The mechanisms of adsorption and desorption are commonly visualized by chemists as individual molecules approaching and interacting with individual surface structures. This approach can lead to simple and understandable models of adsorption processes as, for example, in the work of deBoer (1). It is an easy step from this visualization to defining a formal model which can be simulated by a computer. A discrete event simulation, however, requires the most work out of a computer and, therefore, the most care in the design of the simulation algorithms.

Discrete Event Algorithms

A computer cannot possibly simulate the behavior of a chromatographic experiment in every detail. The most fundamental limitation is the fact that a digital computer is a serial execution machine, while the real world is parallel. That is, the molecules in a real chromatographic column are all being processed simultaneously, but the computer can simulate only one event at a time. A second problem is the very large numbers of discrete events involved. No computer is fast enough or has a large enough memory to model a full-scale chromatographic experiment, so the model experiment must be scaled down to the size of the available computer.

The first step in scaling down the model is to decide which details are really of interest and which can be replaced by simpler approximations or left out of the model completely. For example, diffusion of a molecule in the gas phase along the column can usually be adequately approximated by random numbers from a Gaussian distribution whose variance is determined by the diffusion coefficient and the time spent in the gas phase. The amount of time required to simulate the effect of diffusion in this way is much less than what would be required to simulate all of the individual gas phase events which result in diffusion. In some cases, gas phase diffusion may have an insignificant effect on the simulation results and so can be left out entirely.

The total number of molecules can be drastically reduced since, in a discrete event simulation, there are no detection limits or noise to worry about. The statistical significance of the results is the ultimate lower limit. How many molecules must be included in the simulation experiment depends on the form and intended use of the results. A single average number, such as retention time, can be accurately determined with fewer molecules than the peak shape or other distributions. Also, less precision is required when exploring the qualitative behavior of a model over a range of parameters than for quantitative measurements. As the models being simulated become more complex, the number of molecules must be increased to produce statistically significant numbers of all kinds of events. Sufficient numbers must be included to ensure that the rarest kind of event of any importance occurs enough times to be significant.

Discrete event simulation programs tend to execute certain key routines repeatedly. For gas-solid chromatography simulation, these would include the adsorption and desorption event routines. The efficiency of these key routines can make a significant difference in the overall efficiency of the simulation. But, as is generally true in computer programming, the only kind of efficiency really worth striving for is in the design of the algorithms. Saving a little bit in clever coding of the algorithms is not likely to make more than a few percent difference in speed and practically none in memory space, while the design of an algorithm could make the diffe~ence between a simulation which is practical and one which will not fit into the computer at all.

Random Numbers

The key routine in a discrete event simulation is the random number generator. Decisions are made on what events are to occur during a particular simulation based on the values of random numbers from various distributions. A single event such as adsorption of a molecule on a site requires at least two or three, and in some cases as many as ten or more, elementary random numbers, depending on the algorithms used to simulate the event. Thus, the efficiency of the random number generator is important for the overall efficiency of the discrete event simulation.

It is also critically important to have good quality random numbers. An unsuspected statistical bias in the random number sequence driving a simulation can cause hidden biases in the results of a simulation. A specific sequence is considered to be random only if it passes the statistical test of randomness appropriate for the given distribution (92). Sequences from some sources may appear to be random and even pass many of the simpler tests and still have biases which might cause problems if the sequences are used in ways sensitive to the particular biases.

There are two kinds of sources for random number sequences, each of which has its advantages and disadvantages. A hardware device to generate and digitize electronic noise should produce truly random numbers. But there is always a small doubt whether the device is working correctly or not. It would be very easy for it to pick up a small outside signal and slightly bias the random numbers being produced. The problem would not be detectable except by the proper statistical testing and so could easily go unnoticed. Another problem in using such a device is the considerable amount of time that can be lost in reading numbers into a computer program.

The second and more common way of generating a sequence of random numbers is through the use of a computer algorithm which digitally computes a sequence of numbers which appears to be random; that is, a sequence which passes the statistical tests for randomness. Since the sequence is actually a deterministic sequence and, therefore, not random in the true sense of the word, it is usually called a pseudorandom sequence. But, unless the simulation program contains an algorithm related to the pseudorandom algorithm, it will not be able to tell the difference between the pseudorandom sequence and a truly random sequence.

Random numbers from several different distributions are needed for the discrete event simulation of chromatographic processes. The basic one from which all the others are generated is the uniform distribution in which any number within the range from zero up to, but not including, one has an equal probability of being produced. The algorithm commonly used by standard random number subroutines for generating samples from the uniform distribution is the linear congruential method (93). Pseudorandom numbers from linear congruential algorithms, when taken in groups, are not, however, statistically independent (94). Only a very small fraction of the possible combinations of numbers can be generated and these are related to each other in multidimensional spaces.

There are now better algorithms available, especially for sequences from which the pseudorandom numbers will be taken in groups (94,95). These feedback shift register algorithms can also produce arbitrarily long, period sequences independent of the computer's word size and are faster than linear congruential algorithms. A 16-bit, parallel feedback shift register pseudorandom number algorithm with a 2^{127} period has been implemented in microcode for the HP 2100 computer. A listing of this essential program is given in the Appendix. The required primitive trinomial, $x^{127} + x^{30} + 1$, was taken from the table compiled by Watson (96). The algorithm, as implemented for the HP 2100, requires a table of 128 words in core memory, 17 words of control store memory, and can generate a 16-bit, uniformly distributed, pseudorandom number if 8.86 microseconds.

The exponential distribution is second only to the uniform distribution in importance for discrete models of chromatographic behavior. If a given kind of event has a constant probability of occurring, then the waiting times between individual events of this kind will be exponentially distributed. The classic example of exponentially distributed waiting times resulting from a first-order reaction is radioactive decay. In most cases, molecules desorbing from a surface should follow the same kind of first-order kinetics and have exponentially distributed waiting times between desorptions.

By the central limit theorem (97), a normally (Gaussian) distributed movement results when a large number of independent movements with a common distribution occur sequentially. In a discrete event model, such a sequence of events may be simulated by a single normally distributed random number. Most varieties of molecular diffusion fit into this category of processes.

Pseudorandom number algorithms taken from Ahrens and Dieter (98) have been implemented for sampling from the standard exponential and normal distributions. The exponential algorithm is a slightly modified version of their algorithm SA and produces a floating point result in an average of 370 microseconds. The normal algorithm is their algorithm CT and produces a floating point result in an average of 870 microseconds. Both use uniformly distributed pseudorandom numbers produced by the feedback shift register algorithm. Sequences from these two algorithms were tested for randomness using the Kolmogorov-Smirnov test as described by Knuth (99).

Adsorption Processes

Adsorption and desorption are the most basic events of a gas-solid chromatography discrete event simulation. Figure 3.3 illustrates how these two events are commonly though of in informal kinetic models of chromatographic processes. A molecule in the gas phase flows at the carrier gas velocity until it happens to encounter the surface. It then becomes adsorbed and remains stationary for awhile. Sometime later it desorbs and again moves with the carrier gas.

For a molecule to become adsorbed, it must first encounter the surface through diffusion or flow processes. Regardless of the details involved, this should occur with some average rate with approximately exponentially distributed waiting times. The probability of encountering the surface is higher immediately after a desorption event, resulting in a small bias toward shorter waiting times. In most cases, the qualitative behavior of the model will not be changed significantly although the numerical value of the rates may be affected. Each time the molecule is at the surface it has a chance of becoming adsorbed, depending on its orientation with respect to the surface, the availability of an adsorption site, the possibility of the site already being occupied, and other conditions. If all these possibilities have constant probabilities, then there is no need for the simulation to explicitly consider them. They can all be combined with the average rate of encountering the surface to give an average effective adsorption rate. In the simplest kind of

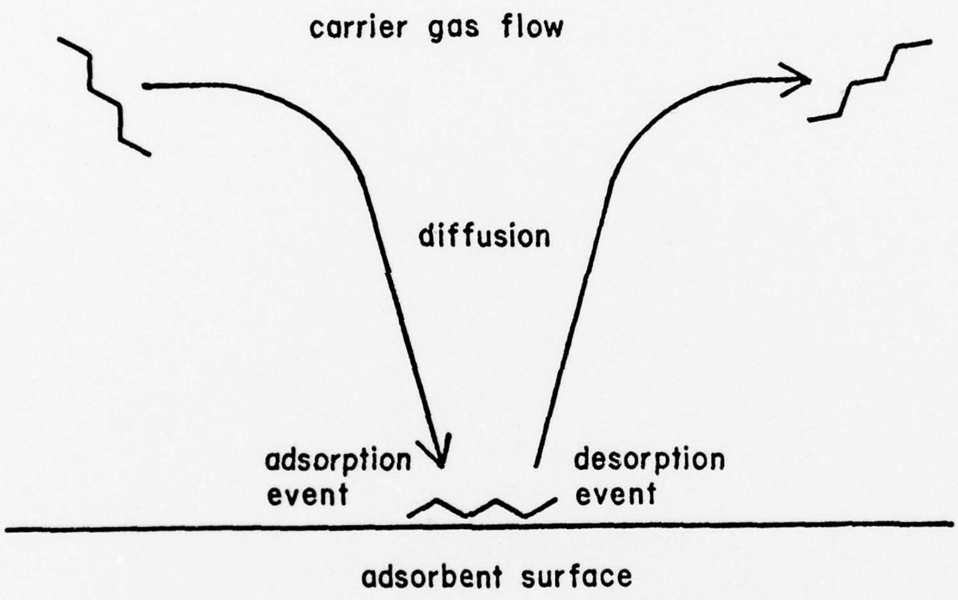
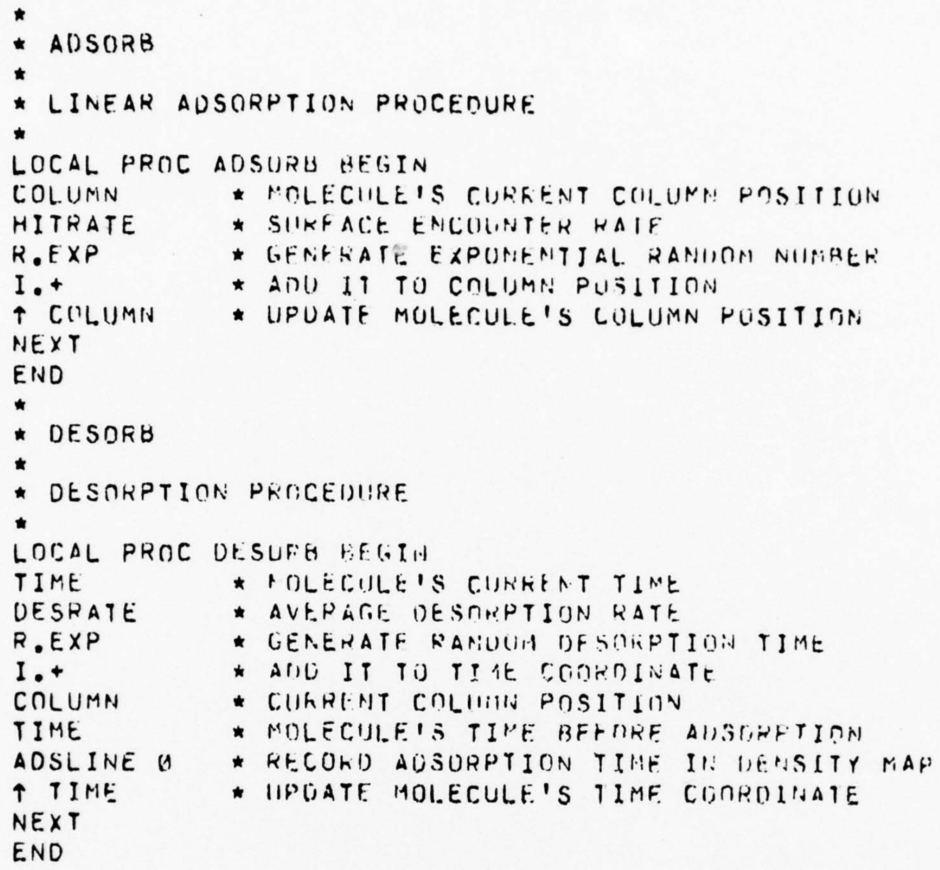


Figure 3.3 An informal model of adsorption and desorption events.

linear adsorption process the whole complex chain of events can be replaced by a single event with an average waiting time between events equal to the sum of the individual average waiting times. Computer models for simple linear adsorption and desorption in a chromatographic column are given in Figure 3.4.

In a linear chromatographic model, the individual adsorption sites do not have to be explicitly included as they are in the adsorption models of Nakagawa (88,89). A linear isotherm implies that there is no competition between molecules for adsorption sites and, therefore, it is sufficient to know the number of sites per length of column. From this an average waiting time between site encounters can be calculated. The column position at which a molecule becomes adsorbed is determined, but the identity of the individual adsorption site is not.

Several major simplifying approximations have been made in the ADSORB procedure given in Figure 3.4. First, diffusion in the gas phase has been ignored. If it is important in a particular model, then simple diffusion could be included by generating random numbers from the appropriate normal distribution and adding them to the column position at each adsorption event. Second, the variation in gas flow rate due to the pressure drop along the column has been left out of the model. It too could be included by making the adsorption waiting time a function of the column position. Third, to reduce the number of calculations involved in the simulation, the model assumes that molecules spend no time in the gas phase. This is equivalent to assuming a zero retention volume for an inert gas sample. The effect can be compensated for by adjusting all



```
*
* ADSORB
*
* LINEAR ADSORPTION PROCEDURE
*
LOCAL PROC ADSORB BEGIN
COLUMN       * MOLECULE'S CURRENT COLUMN POSITION
HITRATE      * SURFACE ENCOUNTER RATE
R.EXP        * GENERATE EXPONENTIAL RANDOM NUMBER
I.+          * ADD IT TO COLUMN POSITION
↑ COLUMN     * UPDATE MOLECULE'S COLUMN POSITION
NEXT
END
*
* DESORB
*
* DESORPTION PROCEDURE
*
LOCAL PROC DESORB BEGIN
TIME         * MOLECULE'S CURRENT TIME
DESRATE      * AVERAGE DESORPTION RATE
R.EXP        * GENERATE RANDOM DESORPTION TIME
I.+          * ADD IT TO TIME COORDINATE
COLUMN       * CURRENT COLUMN POSITION
TIME         * MOLECULE'S TIME BEFORE ADSORPTION
ADSLINE 0    * RECORD ADSORPTION TIME IN DENSITY MAP
↑ TIME       * UPDATE MOLECULE'S TIME COORDINATE
NEXT
END
```

Figure 3.4 Linear adsorption and desorption models.

retention times and volumes by a constant amount after the simulation is complete.

In procedure DESORB, an exponential random number whose average value is determined by the desorption rate is added to the current simulation time for the molecule. The time during which the molecule was adsorbed is then recorded in the molecule density map.

Of course, more is involved in a gas-solid chromatography experiment than just the adsorption and desorption processes. So more is required to make a complete model of chromatography out of the ADSORB and DESORB procedures given in Figure 3.4. As a minimum, the model must provide for molecules being injected into a column, flowing through the column while undergoing the adsorption and desorption, and being detected as they reach the end of the column to produce a chromatogram. A flow chart modelling the behavior of a molecule in a gas-solid chromatographic system is given in Figure 3.5. The same model translated to a computer-understandable procedure is given in Figure 3.6.

At the beginning of the procedure, the molecule is injected into the column. How to do this is specified by another procedure, named INJECT, which may be as simple or complex as desired. In the simplest case, ideal impulse injection, each molecule starts at column position 0 and simulated time 0. All this version of INJECT has to do is set the corresponding variables in the simulation to zero. Any other kinds of injection may be modelled by using an appropriate INJECT procedure.

Figure 3.7 is a procedure to model an exponential decay injection input profile. The width of the profile is determined by an average

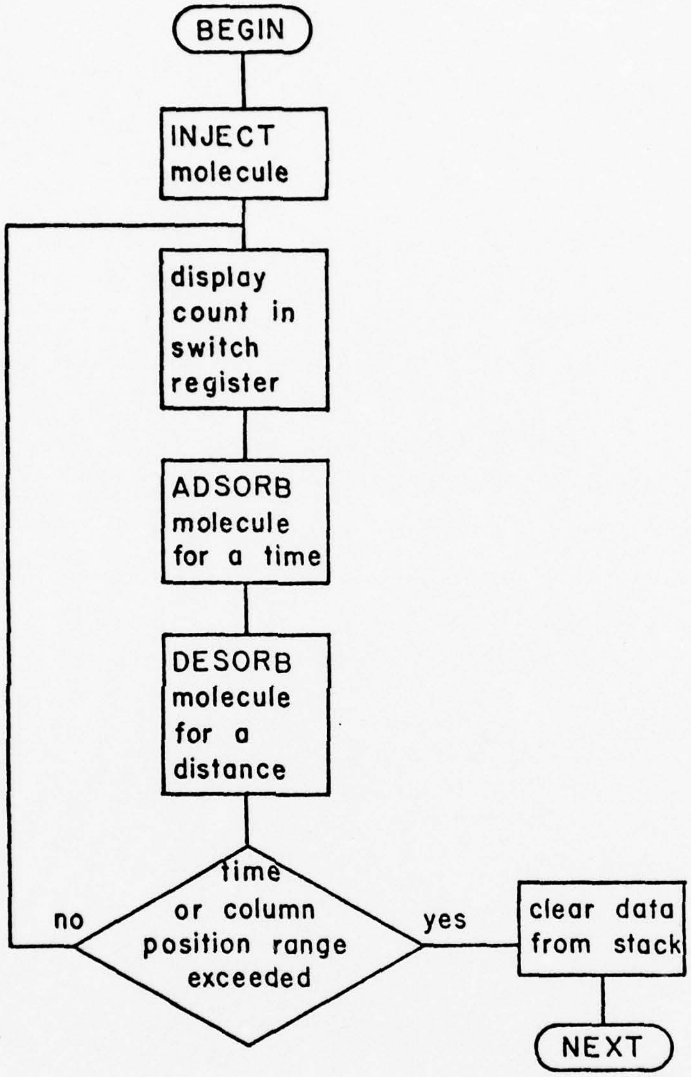
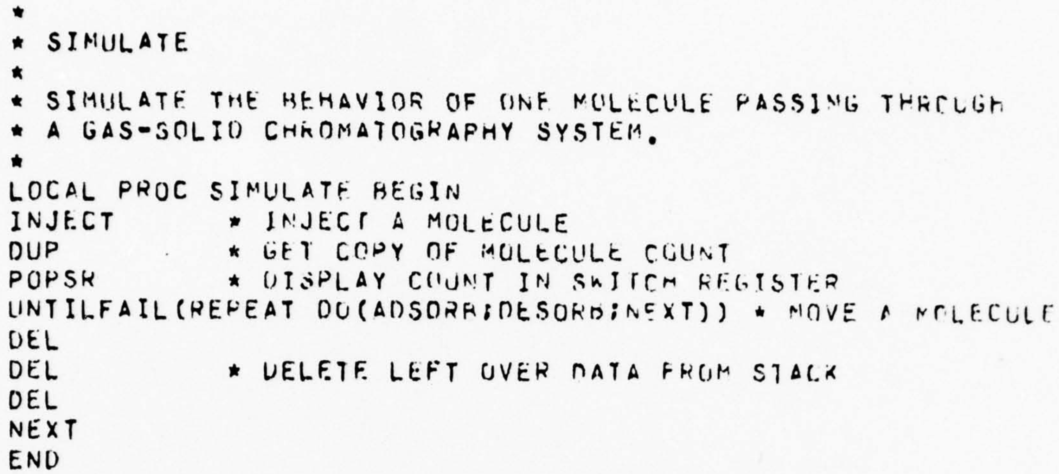

BEGIN
INJECT molecule
display count in switch register
ADSORB molecule for a time
DESORB molecule for a distance
time or column position range exceeded
no
yes
clear data from stack
NEXT

Figure 3.5 Flow chart of a gas-solid chromatography model.



```
*
* SIMULATE
*
* SIMULATE THE BEHAVIOR OF ONE MOLECULE PASSING THROUGH
* A GAS-SOLID CHROMATOGRAPHY SYSTEM.
*
LOCAL PROC SIMULATE BEGIN
INJECT         * INJECT A MOLECULE
DUP            * GET COPY OF MOLECULE COUNT
POPSR          * DISPLAY COUNT IN SWITCH REGISTER
UNTILFAIL(REPEAT DO(ADSORB;DESORB;NEXT)) * MOVE A MOLECULE
DEL
DEL            * DELETE LEFT OVER DATA FROM STACK
DEL
NEXT
END
```

Figure 3.6 Model of a chromatographic process.

```
*
* INJECT
*
* SET MOLECULE COORDINATES TO INITIAL VALUES
*
LOCAL PROC INJECT BEGIN
INJRATE        * INJECTION RATE PARAMETER
R.EXP          * GENERATE EXPONENTIAL INJECTION DELAY
↑ TIME         * TIME A WHICH MOLECULE ENTERS COLUMN
I. Ø           * PUT A ZERO ON DATA STACK
↑ COLUMN       * MOLECULE ENTERS COLUMN AT BEGINNING
NEXT
END
```

Figure 3.7 Molecule injection model with injection port mixing.

30

injection rate parameter, INJRATE. This type of profile should result from a plug injection into an injection port with either a significant mixing volume or sample adsorption on injection port and connecting tubing walls. All of the simulations presented here use this model, usually with a fairly high injection rate parameter so that the injection profile will have little effect on the simulation results. Completely ideal injections should not be used with these simulation algorithms because all molecules would then have to pass through a single point. This single point i: a mathematical singularity and cannot be adequately represented by a finite precision computer memory.

Once a molecule is injected into the column, it is repeatedly adsorbed and desorbed as specified by the ADSORB and DESORB procedures. When either the simulation column position or time exceeds the range allowed for the simulation, then an outside-of-range condition is detected and the computer exits from the adsorb-desorb loop.

The procedure SIMULATE, given in Figure 3.6, models the behavior of a single molecule in a gas-solid chromatography experiment. The details of the experiment depend upon the definition of the submodels INJECT, ADSORB, and DESORB, along with any parameter values used by them. SIMULATE is in turn a submodel for some other procedure which determines the values of any required parameters, executes SIMULATE for each molecule in the experiment, and presents the results of the experiment. For statistical significance, a large number of molecules must be simulated, the minimum number depending on the required precision. The best way to simulate many molecules would be to provide many processors, one for each molecule, all executing the procedure SIMULATE in synchronization. The approach is possible, although expensive now. But with the rapidly falling prices for simple microprocessors, it may some day be possible to run discrete event simulations in parallel with at least a moderate number of molecules. With the present single processor computer, however, some method must be employed to simulate the parallel operation of many molecules while actually operating serially.

The computer could execute the events in their correct order, switching around between different molecules as required, or it could run a single molecule through the complete simulation before starting another one. The first technique is the more realistic and is the way most discrete event simulations must be run in order to observe effects due to the interactions between events, but it is also very inefficient due to the record keeping required in switching around between molecules. The second technique is more efficient, easier to program, and results in more understandable programs and so should be used whenever possible. In linear chromatography models, there are no interactions between molecules anyway, so the event sequence can be rearranged to better suit the computer. The event sequence in some nonlinear chromatography models may also be rearranged if some other means of simulating the interactions between molecules is provided.

Molecule Density Maps

The time at which each molecule emerges from the column is a very important simulation result because it is directly comparable to the results of a real experiment, e.g., the chromatogram. A simulation program, however, is not limited in the same ways the real system is. Information can be collected from a chromatographic column only through an appropriate detector attached to the end of a column. There is no convenient way to observe the development of a peak as it moves down the column. In a simulation, however, the column position and time of each adsorption and desorption event is known so statistical records can be kept of the simulation's behavior at any point in the experiment. For example, the times at which molecules cross a series of points spaced along the column ma) be recorded producing, in effect, a series of chromatograms for various length columns. Or, the column positions of all molecules at certain points in time may be recorded to produce plots of molecule density along the column at specific times. Combining and generalizing these two ways of recording additional data leads to a two-dimensional representation for simulation results in which the x-axis is column position and the y-axis is time. Each molecule of the simulation then has a trajectory in this two-dimensional space, moving in the x-direction while in the gas phase and in the y-direction while adsorbed on the surface. Figure 3.8 illustrates a typical molecule's trajectory in this two-dimensional representation. Statistics are collected from a simulation by recording the number of molecules passing through each point on the plane. If the density of molecules passing through is plotted vs. column position and time in three dimensions, a graph similar to Figure 3.9 is produced, illustrating the detailed development of a peak as it moves down the column with passing time. A slice through this density map at a given column position is the chromatogram which would be

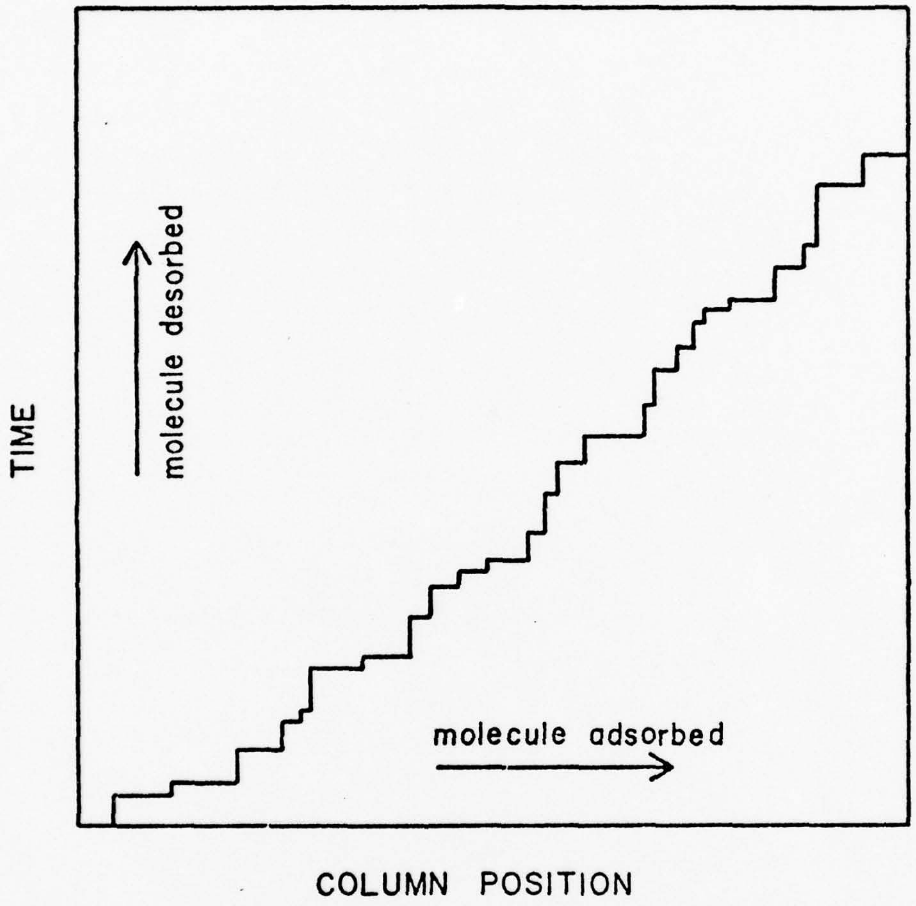


Figure 3.8 Trajectory of a typical molecule through a density map.

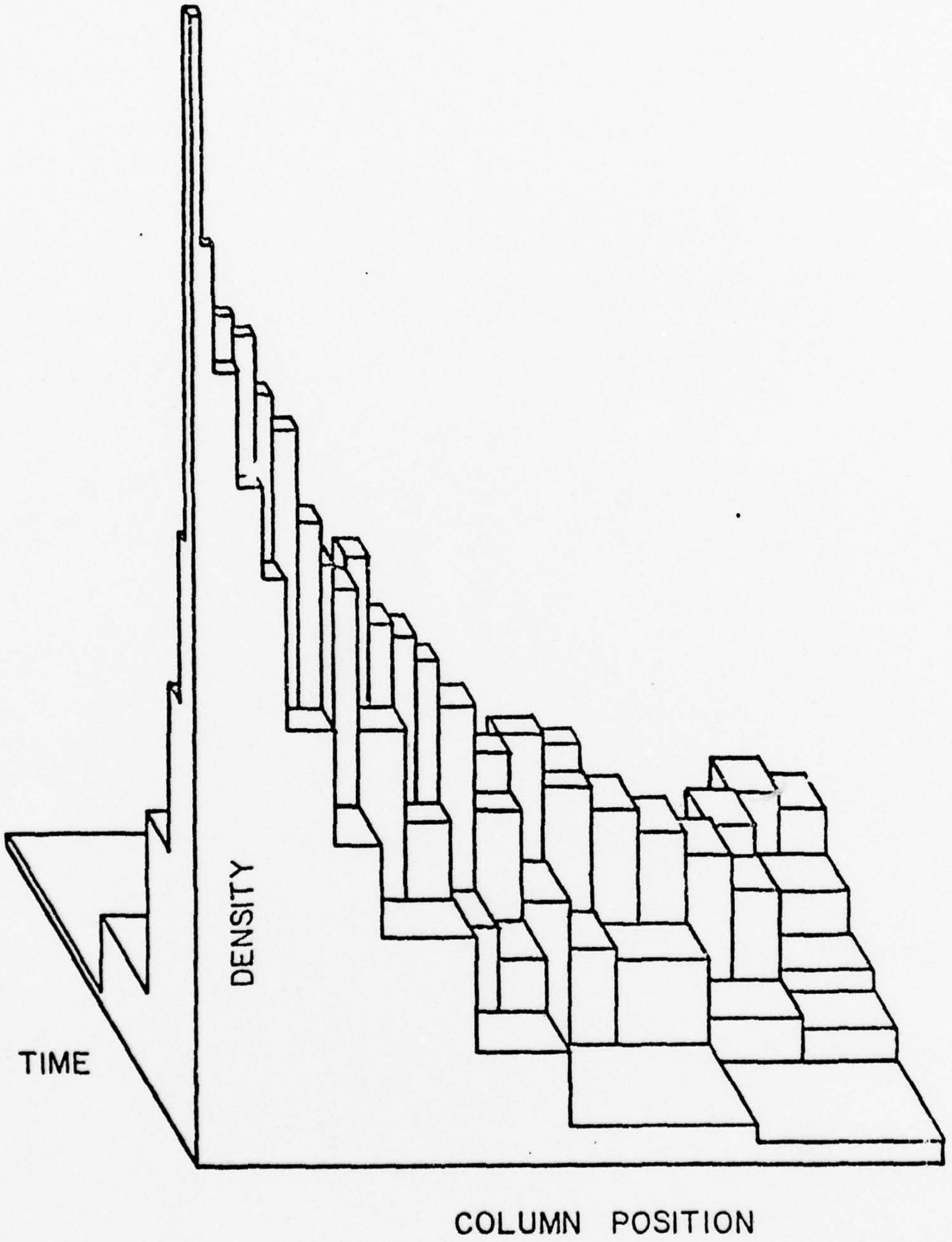


Figure 3.9 A molecule density map.

recorded by an ideal detector at the end of a column of that length. A perpendicular slice at a given time is the molecule density along the column plot for that time.

These molecule position and time density maps provide a means of simulating nonlinear adsorption processes in a chromatographic column without running all the molecules simultaneously. The behavior of each molecule can be simulated from injection to the end of the experiment without being concerned with any other individual molecules just as in a linear simulation. The decision on whether a molecule is to be adsorbed or not at a given point along the column depends on the probability of the molecule encountering the surface at that point and on the density of available sites at that point at the time the molecule is there. If it is assumed that there is no significant interaction between molecules in the gas phase, then encountering the surface is a first-order linear process and the presence of other molecules does not affect its probability of occurring. The probability of actually changing to an adsorbed state, however, depends on the density of molecules already adsorbed. In the case of a Langmuir isotherm, the molecule can become adsorbed only if it encounters an empty adsorption site. This is a nonlinear process because the probability of an empty site being available is determined by the number of molecules already adsorbed. It is required that the density of adsorbed molecules at the proposed adsorption site position and time be known, but not that the individual identities of the adsorbed molecules be known. The molecule density maps provide just this information.

A nonlinear adsorption procedure for gas-solid chromatography simulation is given in Figure 3.10. In procedure ADSORB, the adsorption process is broken up into two steps. First, the molecule being simulated moves down the column until it encounters the surface in procedure HITSURF. The distance moved is sampled from the average surface encounter rate exponential distribution. The procedure STUCK determines the density of molecules at the current column position and time and, using a uniform distribution random number, makes a decision on whether the molecule will become adsorbed or not. If the density of molecules is high at this point, then the new molecule is not likely to find an available adsorption site and so will remain in the gas phase and HITSURF must try again. If the density of molecules is low, then the new molecule will most likely become adsorbed. The desorb procedure is identical to the one used with the linear adsorption in Figure 3.4.

Information provided by a molecule density map is incomplete because it includes only information about molecules which have already been run through the simulation. The trajectories of molecules yet to be simulated cannot affect the behavior of those already done. This is an inescapable consequence of reordering the sequence of events. The reordering is necessary to make this discrete event simulation of gas-solid chromatography practical, but it also affects the simulated behavior of nonlinear models. As the number of molecules simulated increases, the information provided by a molecule density map should become more accurate. To attain a given level of accuracy, more molecules must be simulated if the sequence of events is reordered than if all the molecules are simulated simultaneously. If the final density map converges to the same result as would be attained by a simultaneous simulation, then reordering the sequence of events is worthwhile even if it should require several times as many molecules.

Each different kind of adsorption site in a gas-solid chromatography model must have a density map of its own in the simulation. The different kinds of sites may have different densities and adsorption kinetics so that when one kind of site is nearly full in a region of its density map, another may have room for many more molecules in the same region. Different adsorption and desorption kinetics for different kinds of sites may cause the shape of chromatographic peaks adsorbed on the various kinds of sites to be different. Molecules may go from being adsorbed on one kind of site to the gas phase and then on to any other kind of site. Consequently, the behavior of one density map can affect all the others in complex ways. Because of these unpredictable differences in the density maps for different kinds of sites, there is no way to calculate the local concentration of molecules on a particular kind of site given the total concentration at that point along the column and in time.

The basic routines for handling molecule density maps provide for up to eight independent maps in a single simulation. These can be allocated as needed to simulate the various kinds of adsorption sites and molecules in a particular model.

A molecule density map must be stored as a two-dimensional data structure in the computer's main memory. It must have a definite value at each pair of time and column position coordinates. The coordinates are specified by 15 bit positive integers providing a precision of one part in 32768.

The obvious core memory organization for a density map is the familiar two-dimensional array. This structure, however, is unacceptable because of the excessive amounts of memory required to attain the required precision in the coordinates. An array of dimension 100 by 100 would require 10,000 words of core memory. At most, only two such arrays could be stored in the computer's main memory at one time. In most cases, even arrays of this size would have insufficient coordinate precision.

The problem is not that the computer's main memory is too small to hold all the information in a density map. It is that the array structure is very inefficient in storing this information. In a typical simulation, large portions of a density map may have few if any molecules passing through them. Such areas of the density map do not need the fine coordinate precision or full word range provided by the standard array structure. In all parts of an array density map, there would be considerable correlation between neighboring coordinates so that the actual independent information contained in a word of the array would be much less than that allowed by the 16 bit word size. Thus, a density map stored as a two-dimensional array would contain much redundant information and occupy much more memory space than is actually required.

The one advantage of the array structure over other possible data structures for density maps is its access speed. The number of operations required to index an array is inherently small and, in addition, the instruction sets of most computers are designed to perform these operations efficiently. It is important that whatever structure is used to store density maps in the computer's memory be efficient in access speed as well as memory space. The execution speed of these discrete event simulations of nonlinear gas-solid chromatography is determined more by the access speed of the density maps than by anything else except the generation of random numbers.

A binary tree (100) structure was chosen for representing density maps in the computer's memory. Each of the eight density map binary trees is accessed beginning at a root node contained in an eight element array. The root node contains two pointers to subtrees each representing half of the density map. Initially, each of these two subtrees contains but two equal terminal nodes. The density map as a whole is thus split into four equal parts with the same density of molecules in each one and the column position and time coordinates each having only one bit precision. During execution, the trajectories of molecules are recorded in the density maps to the precision currently allowed by the binary tree by incrementing the value of terminal nodes along adsorbed portions of the simulated molecule's trajectory. Whenever a terminal node exceeds its maximum value of 131, it is split into two new terminal nodes, each of which begins with half the total counts from the splitting node. The counter in a former terminal node which has been split is replaced with a pointer to the new pair of terminal nodes, its descendants. As the total number of counts recorded in a density map during a simulation increases, a scale factor is also incremented in such a way that the total density of molecules in the density map remains constant. Executing the simulation changes the distribution of molecule density, but does not change the total density. Figure 3.11 illustrates how a molecule density map is divided up into smaller units in regions of higher density. Each rectangle is represented in the computer's memory by a terminal node of the binary tree and is accessed by following the tree structure from the root node. In Figure 3.9, the third dimension, molecule density, has been added.

As a simulation is executed, the total amount of memory space occupied by a density map increases. Eventually all of the memory will be occupied. Simpler simulations, especially those that are not too far from linear, can converge to an acceptable result before the memory space is exhausted. In many cases, however, the simulation may use all of its available memory before it has converged to a final result. If it has not converged because of insufficient precision, then nothing but more memory will help. If the precision is adequate but it just has not stabilized yet, then more memory may be obtained by pruning the binary tree structure. A binary tree is pruned by uniformly reducing the value of each terminal node by 1/4 and recombining any adjacent terminal nodes with values both less than 68. The scale factor for the binary tree is adjusted by the same amount so that nothing is changed except that more memory is available to simulate more molecules and the current precision is reduced. Since it is the results of the earlier molecules which are pruned away, this has the effect of giving more emphasis to later

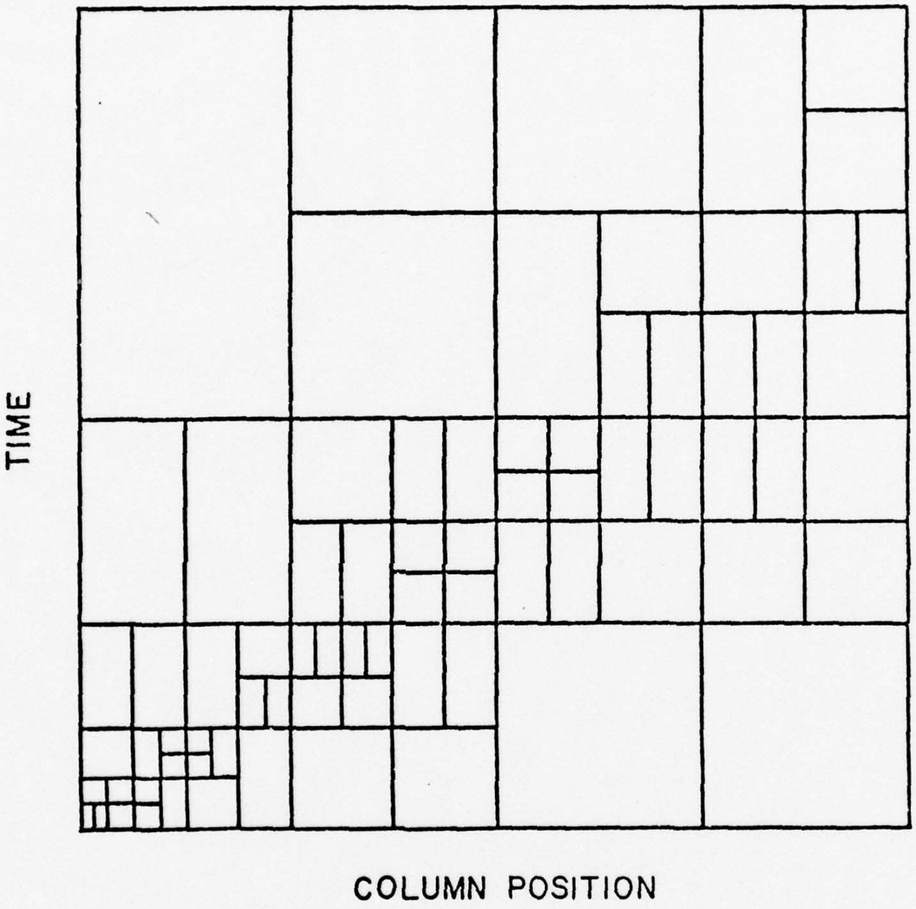


Figure 3.11 A binary tree partitioning of a molecule density map.

molecules in the simulation which should have more accurate behavior. Thus, if the overall precision is adequate, a periodic pruning of the binary tree structure should result in a faster convergence to the final result as well as allowing simulation to continue after the available memory has been exhausted.

A simulation is assumed to have converged to a final result when the density maps are no longer changing, except for minor statistical fluctuations. Sharp fronts on nonlinear peaks are usually the last parts to become stable. If there is insufficient memory space, then these sharp fronts go through repetitive cycles, often with extra peaks forming and decaying without ever reaching a stable result. All simulation results presented in this work have stabilized for at least the last half of the molecules simulated.

For models with a significant amount of nonlinear behavior, density map convergence requires many more molecules than are needed for statistical significance. It is usually the amount of information that can be stored in memory rather than the total number of molecules or events simulated which limits the accuracy of a simulation. Once memory space is exhausted and the binary trees are pruned, then additional simulated events cannot contribute to statistical significance because there is no more room to store any more precision. Convergence, however, can continue.

To determine the density of molecules at a particular point in a density map, the simulation specifies the coordinates of the point with the full allowed 15 bit precision. Then, starting at the appropriate root node, the density map access subprogram searches the binary tree, making a decision on which way to go at each split node based on the value of bits from the supplied coordinates. When a terminal node is finally reached, the value of the density map at this point is calculated from the number of counts in the terminal node, the number of splits nodes that are passed in reaching it, and the current value of the scale factor. Each terminal node value is recorded with seven bits of precision.

To record the trajectory of a molecule in a density map, the simulation specifies the end point coordinates for a line segment part of a trajectory with the full 15 bit precision allowed. Then, starting at the appropriate root node, the density map access subprogram searches the binary tree in the same way it did for determining the value at a point. But when a terminal node is reached, that node is incremented and possibly split into two and the scale factor is adjusted. Then the initial coordinate of the line segment being recorded is incremented by an amount specified for this particular binary tree. If the initial coordinate is still less than the line segment's final coordinate, then the binary tree search and increment procedure is repeated.

As the simulation is executed, more nodes are split, producing more branches in the binary tree structure and allowing greater precision in the use of time and column position coordinates. Those parts of the molecule density maps with the greatest density are also the parts with the greatest coordinate precision. The regions through which few, if any, simulated molecules ever pass have very poor coordinate precision. The precision to which the coordinates can be specified in a given region of a density map during a simulation is proportional to the amount of information which has been used to create the structure of that region. If only a few molecules have passed through a region, then the details of the distribution of molecules in that region cannot be very well known anyway and there is no need to specify the coordinates with high precision. In fact, specifying them with higher precision than allowed by this binary tree structured density map is less accurate on the average because of statistical variations.

A terminal node can have a value between 4 and 31 inclusively. When it reaches 132, it is split into two descendent nodes each of which begins with 66 counts. Thus, at any point in a density map, the precision of the calculated value is limited to 7 bits. During execution, the increase in precision goes into further dividing the time and column position coordinates rather than into the precision of individual points. Contrast this with the array data structure in which the precision of individual points increases while the precision of the coordinates is fixed. Besides being much more efficient in the use of core memory than an array, the binary tree structure also more accurately represents the information which a density map is supposed to store. As a result, fewer molecules should be required for the density map to converge to a true representation of the distribution of molecules in a nonlinear gas-solid chromatography simulation.

The major deficiency of the binary tree structure for density maps is its access speed. To reach a terminal node, the program must fetch and examine each split node leading to it. In contrast, a point in an array structure can be accessed using at most one multiply and two add instructions. This disadvantage can be overcome somewhat through the use of the HP 2100 computer's microprogramming capabilities. But still, the binary tree structure must be searched and this is inherently a less efficient procedure than indexing an array. At the beginning of a simulation before the binary tree density maps have grown very much there should be little difference in speed. The binary tree structure may even be a bit faster because there are fewer counters to increment in recording a molecule's trajectory. But as the simulation runs and more nodes are created, the access speed will begin to slow down. Also, as the density map converges to the final result, the majority of molecules will follow an average trajectory through the densest parts of the density maps which are also the most finely divided and the slowest to access. The advantages of the binary tree structure are, however, so great that this one disadvantage is insignificant in comparison.

In terms of memory efficiency, this binary tree structure for storing molecule density maps is much better than a two-dimensional array, but it is by no means the most efficient possible. Any one of a great many functional forms, for example, splines or two-dimensional polynomials, could be used to store the information in the computer's memory. Most of these, however, are unacceptably slow in recording molecule trajectories and calculating the density of molecules at a point.

Simulation Scaling

The molecule density maps upon which this discrete event simulation is built are implemented using dimensionless numbers for all parameters and results. The ranges of the numbers used are determined by the structure of the computer's memory and instruction set rather than the chromatographic experiments to be simulated. This was done for two reasons. First, the computer can process information much more efficiently if it fits into its own scheme of things rather than in some arbitrary outside o.ganization. This is the same reason the density map structure was used. Second, and just as important, the use of dimensionless numbers allows much greater flexibility in the range of real chromatographic models which can be simulated by a single simulation program.

All parameters and results used in this study are specified or presented in terms of the dimensionless internal numbers. They could be scaled to match real chromatographic systems, but the main purpose of these first simulations is to test and demonstrate the workings of the simulation itself and to develop a qualitative understanding of nonlinear mechanisms in chromatography rather than to derive values for parameters in specific chromatographic models. Scaling the parameters would make little difference in the ease with which the simulation could be used because the main objects of study, nonlinear mechanisms, are the same regardless. Implementing models using scaled parameters would result in significantly reduced efficiency because parameters would have to be continually translated to internal dimensionless form during simulation. Scaling parameters, when they are input to the model, would restore the simulation efficiency but the user would be required to supply the appropriate scaling algorithms whenever he changes a model. An understanding of the internal dimensionless form is still required of the user plus the work involved in writing scaling routines.

The simulation's basic internal dimensions are the molecule density map coordinates in both time and column position and, on an independent scale, the density of molecules at any point in the map. Time and column position coordinates each have 15 bits precision and, therefore, a dimensionless range of 0 to 32,767. The last bit of the 16 bit work is used by the basic density map access routine to detect a coordinate overflow or out-of-range condition which occurs whenever a molecule reaches the simulated column or time limit. The density of molecules at a given point in the density map is stored in the binary tree structure as discussed previously. The basic access routine to find this density at a given pair of coordinates calculates a density from the binary tree structure and returns it as a standard format floating point number. The actual scale of this number has been arbitrarily chosen to be one. Before any molecules have been run through a simulation, the density at all points in the density map is exactly one. As the simulation is executed, molecule density is added to some regions of the density map but not to others, resulting in some regions increasing in density above one and others dropping below and approaching zero. All other internal parameters for a simulation are in some way related to either the density map coordinates or total density and must be scaled to match them.

A discrete event model is necessarily a kinetic model and therefore requires reaction rates as parameters. The two rates of interest in these models are the surface encounter rate and the desorption rate. The internal scale of the surface encounter rate is determined by the column position coordinate because the molecule is moving in that dimension while it is in the gas phase. The parameter as specified is the probability of the molecule encountering the surface at each unit of column length as it passes through. The reciprocal is therefore the average number of column units passed through between surface encounters and the total column length (32,767) multiplied by the parameter is the average number of surface encounters by a molecule passing through the column. The internal scale of the desorption rate is determined by the time coordinate because the molecule is moving in that direction while it is adsorbed. Desorption rate parameters are specified in a form analogous to surface encounter rate parameters and can be interpreted similarly.

The internal concentration or molecule density parameter is a dimensionless ratio, the number of sites per molecule. A typical gas-solid chromatography system might contain 1 g of 30 m^2/g adsorbent packed in a 1 m column for a total surface area of about 30 m^2 (101, 102). A nonspecific adsorbent surface should be able to accommodate about 4 molecules per nm^2. Thus, there should be about 10^{20} adsorption sites in this column. If a peak occupies on the average about 1% of the column at a time, then the total number of sites available is about 10^{18}. A typical sample injection is about 10^{-9} g which at 100 g/mole is 10^{-11} moles or about 10^{13} molecules. Thus, a site per molecule ratio of 10^{5} should give typical linear chromatographic behavior. Increasing the sample size by a factor of 10^{3} gives a sites-per-molecule ratio of 10^{2} and results in nonlinear behavior in both the real chromatographic system and the simulation.

Linear Chromatography Simulation

Figure 3.12 is a slice through a molecule density map at a fixed time. It shows the density of molecules as a function of column position at this time in a linear simulation. This plot of molecule density vs. column position is not all of the information contained in a density map. Plots may be made at other values of the time coordinate to illustrate the development of the peak at earlier or later times in the simulated experiment. In addition, a slice of the density map at a fixed column position may be plotted vs. the time coordinate.

The density map from which Figure 3.12 was taken is the result of a simulation of the model given in Figures 3.5 and 3.6. Linear adsorption was modelled by the ADSORB procedure given in Figure 3.4. The peak matches the expected shape for a linear chromatographic system with slow kinetics. It is not quite symmetrical because at this time in the simulated experiment the average molecule has been adsorbed and desorbed only 32 times. The symmetrical Gaussian peak shape is approached in a chromatographic system only after a long time and a large number of adsorption-desorption events. Most theories of

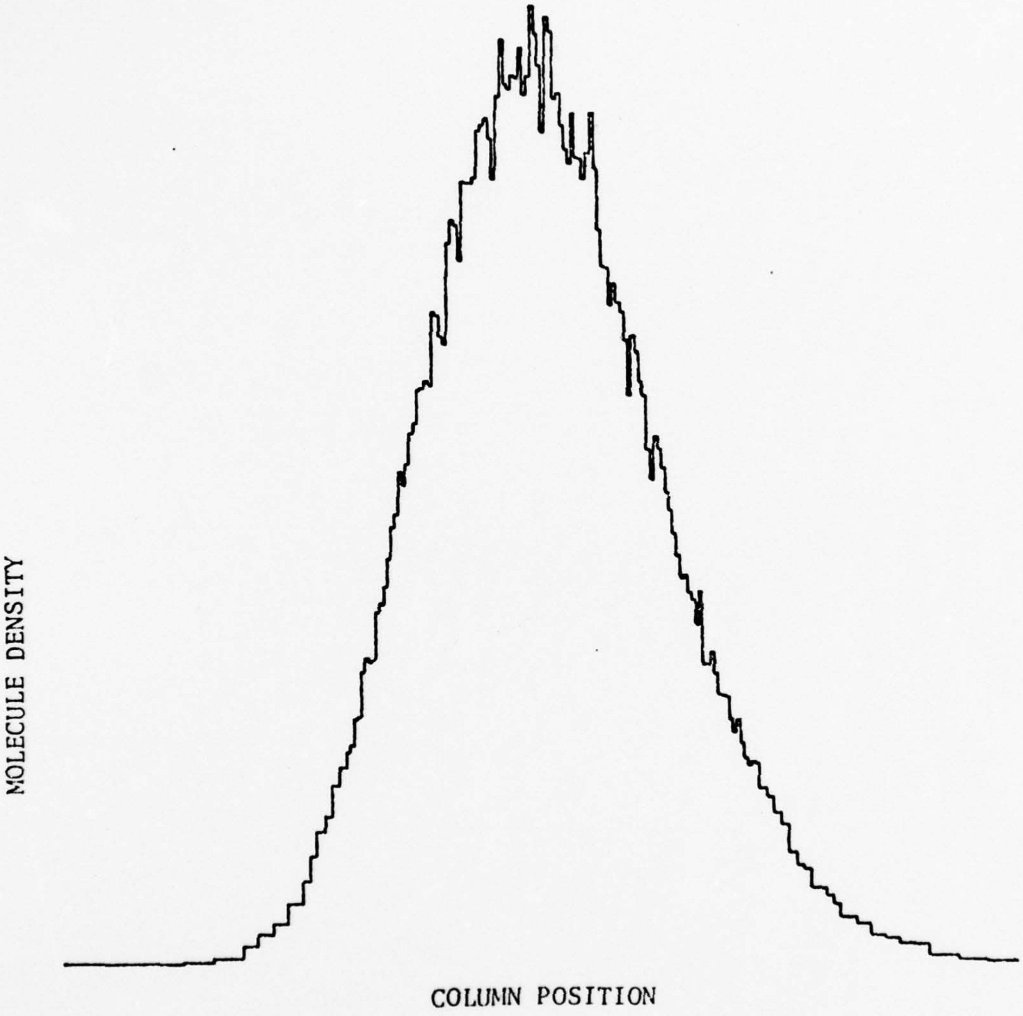


Figure 3.12 Simulation of a linear chromatography model.

Adsorption rate = .002, desorption rate = .002, plot time = 16000.

chromatography must make a long time assumption, but this discrete event simulation actually works best at short times and, therefore, is especially useful in studies of the initial stages of a chromatographic experiment.

Figure 3.13 is an example of a discrete event simulation used to observe the effect of injection profile on a chromatographic peak's initial development. Plot A resulted from a near-ideal injection using the procedure given in Figure 3.7. Plots B and C resulted from progressively broader injection profiles. The front or right side of plots B and C are nearly as sharp as plot A. They are shifted left toward longer retention times simply because the average molecule enters the column later. The tailing side of plot B is also nearly as sharp as plot A. Plot C, however, has an obviously more gentle slope on its tailing side. The overall shape of plot B closely resembles plot A because the injection is sufficiently fast for all of the molecules to get out the injection port and undergo at least several adsorption and desorption events by this time. The overall shape of plot B is determined more by the chromatography than the one time effect of the slow injection. Not all of the molecules in plot C have even entered the column yet so the exponential decay of the injection port still has a large influence on the peak tail.

Nonlinear Chromatography Simulation

The gas-solid chromatography simulation procedure given in Figures 3.5 and 3.6 can be made to model nonlinear behavior by inserting a nonlinear ADSORB procedure into it. A Langmuir adsorption model,

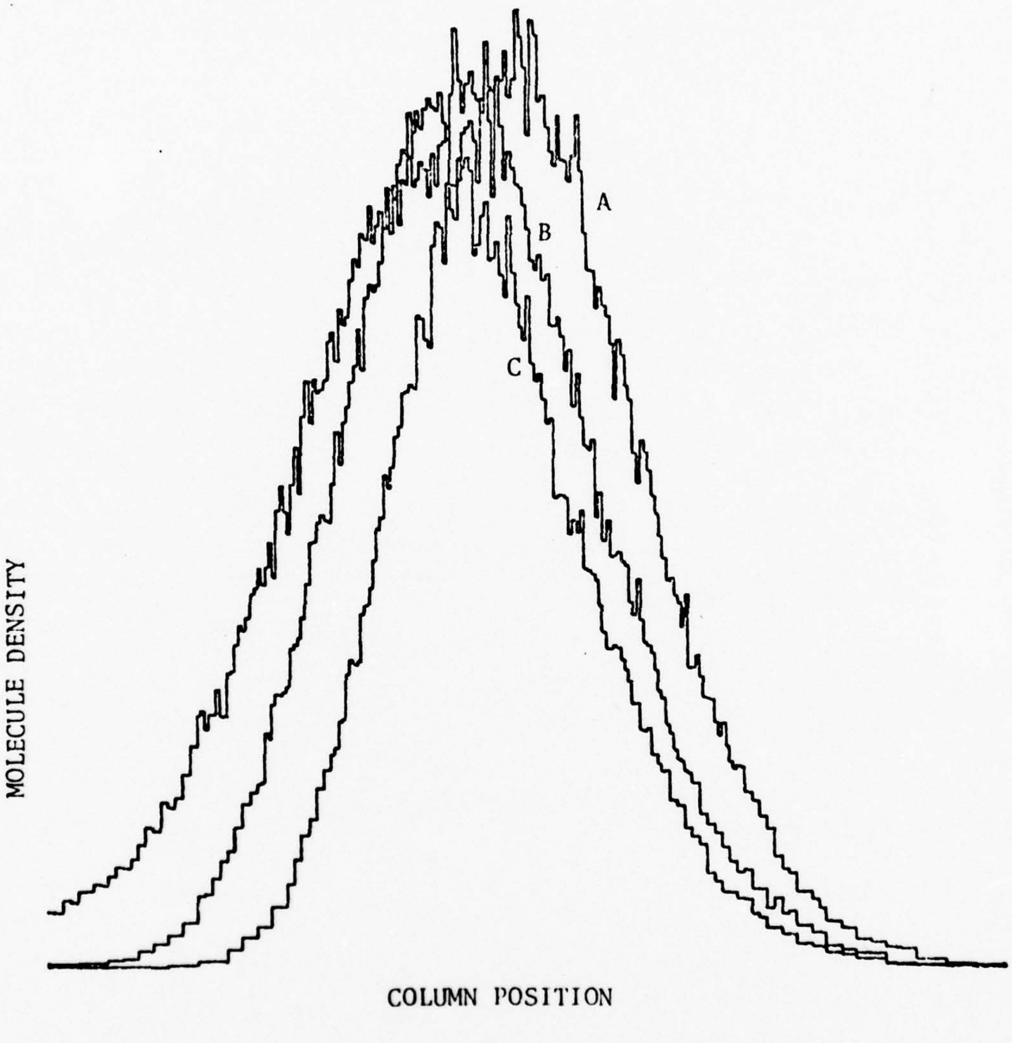


Figure 3.13 Effect on injection port mixing on linear chromatography model.

Adsorption rate = .002, desorption rate = .002, plot time = 16000. Injection rates = 1.0, .0005, and .00025 for plots A, B, and C, respectively.

defined in Figure 3.10, is used in this series of simulations. The rest of the chromatography model remains as it was for the linear simulations. The user can replace any of the modular pieces to build and simulate any kind of chromatographic model.

Nonlinear Peak Development

The results of a simulation of a nonlinear Langmuir chromatographic model are presented in Figure 3.14. Plots A through D show the simulated peak at four points during the simulated experiment. Shortly after injection, plot A, it is still quite sharp but asymmetrical. The front is vertical while the back side is beginning to develop a tail. This asymmetry is the opposite of that observed in a linear model at short retention times (Figure 3.12). At simulation time 4000, plot B, the front has moved a considerable distance down the column but the tail has moved more slowly, resulting in a broader asymmetrical peak. The peak maximum just behind the sharp front is moving at a higher speed than the regions of lower concentration in the tail.

Each successive plot of this developing peak is at a time twice as far along in the simulation as the previous plot. Plot A is at an internal dimensionless time of 4000. Plot B is at 8000, etc. The rate of movement by the peak maximum does not follow this series. During the first 4000 time units, the peak maximum moves approximately 64000 distance units, a speed of about 1.60. But between 4000 and 8000 time units, it moves only about 4600 distance units for a speed of approximately 1.15. By the time the peak reaches plot D, the average speed of the peak maximum is down to 0.69. If this has been a linear model with

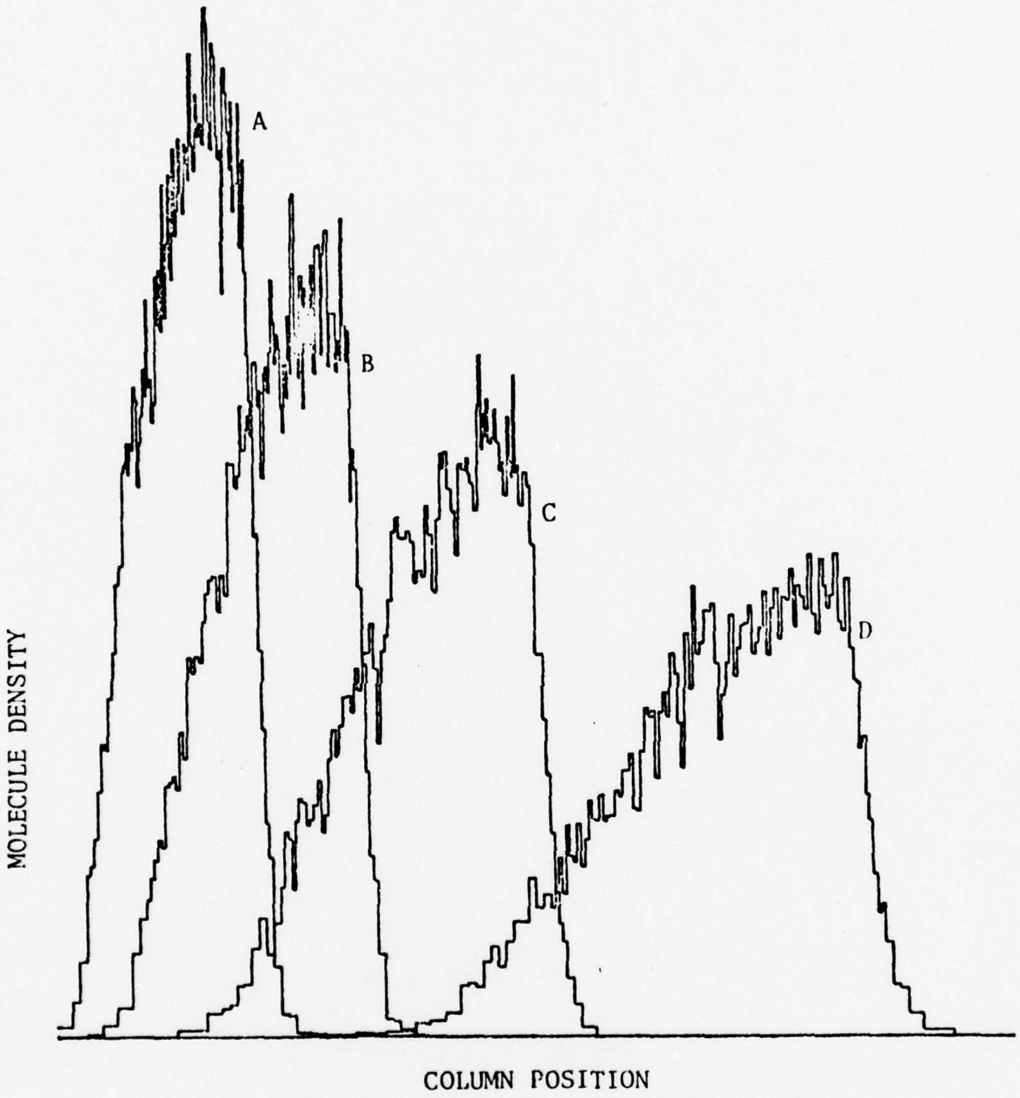


Figure 3.14 Development of a nonlinear chromatographic peak.

Surface encounter rate = .008, desorption rate = .004, sites per molecule = 16. Plot times = 4000, 8000, 16000, and 32000 for plots A, B, C, and D, respectively.

the given rates of adsorption and desorption, then the average molecule would have traveled at a speed of 0.50. The extreme parts of the tail move at about the linear speed, but the rest of this nonlinear peak moves quite a bit faster, resulting in excessive peak broadening. From the point of view of the individual molecules, there is a shortage of adsorption sites. Sometimes when a molecule encounters the surface and tries to become adsorbed, it finds that the adsorption site is already taken so it must remain in the gas phase and continue travelling down the column. This is more likely to happen in regions of high concentration such as exist in the peak soon after injection. So the average molecule and especially those in the part of the peak with greatest concentration move down the column faster than they normally would until the peak spreads out and the concentrations of molecules on the surface is reduced.

The faster moving peak maximum leaves the tail of the peak further behind as it moves down the column. The various parts of the tail move at speeds related to their local concentrations on the surface. Thus, the tail stretches out further as long as any part of the peak contains enough molecules to cause a significant amount of competition for the available adsorption sites. This nonlinear mechanism is a very effective way of broadening a chromatographic peak.

Besides leaving a tail behind, the faster moving peak maximum moves continuously into new parts of the column with completely empty adsorption sites. Those molecules which happen to be in the forefront of the peak encounter this fresh surface and become adsorbed with no competition for adsorption sites. Their speed down the column must be at the slower linear rate for as long as they remain in the lead. But the peak maximum following close behind is moving faster and quickly overtakes them. Any individual molecules which, because of statistical variation in their individual adsorption and desorption behavior, happen to drift ahead of the main body of the peak are slowed down by the availability of adsorption sites and kept close to the peak maximum. The peak cannot spread in the forward direction as a linear chromatography peak would. This self-sharpening front behavior for nonlinear peaks is predicted by other nonlinear theories of chromatography (103,104).

In plots A and B of Figure 3.14, the developing peak still has a self-sharpening front. The slope of the front is not quite vertical but it remains at a constant angle much sharper than the rapidly developing tail. The slope of the self-sharpening front is related to the kinetics of the adsorption and desorption processes. This effect is now shown by the nonlinear chromatography theory of Helfferich and Klein (103) which assumes equilibrium conditions at all times and consequently predicts a vertical self-sharpening front. A nonequilibrium Langmuir isotherm treatment by Zhitomirskii et al. (104) shows nonvertical self-sharpening fronts very similar to these simulation results.

In plot C of Figure 3.14, most of the peak front still has the self-sharpening angle. The concentration at the peak maximum is still high enough to keep it moving faster than any molecules tending to drift to drift ahead. But a few molecules are able to get ahead of it temporarily causing more rounding off at the base of the self-sharpening front.

In plot D of Figure 3.14, the concentration at the peak maximum has decreased further reducing the peak's speed. The angle of the front has fallen below the self-sharpening angle and the front is beginning to spread out as faster molecules drift away. The peak maximum is still moving a little faster than the linear chromatography speed and so can keep the front from spreading quite as fast as it normally would. But, the front is no longer self-sharpening and so much spread out.

Sample Size Dependence

Figure 3.15 presents the results of a series of simulations run with identical parameters except for the size of the injected samples. The smallest peak resulted from an injection whose concentration was small enough to give essentially linear chromatographic behavior. Each successively larger peak resulted from an injection of twice the previous sample concentration. All peaks are plotted at the same point in simulated time and on the same scale relative to the number of available adsorption sites in the column.

The second smallest peak appears to be quite symmetrical and Gaussian in shape. It looks like a linear peak, but its peak maximum is shifted a little toward shorter retention times and it is a little broader than the smallest peak. A truly linear peak should be identical to all smaller peaks except for a scale factor. This second smallest peak is behaving as a linear peak at the time this plot was made in the

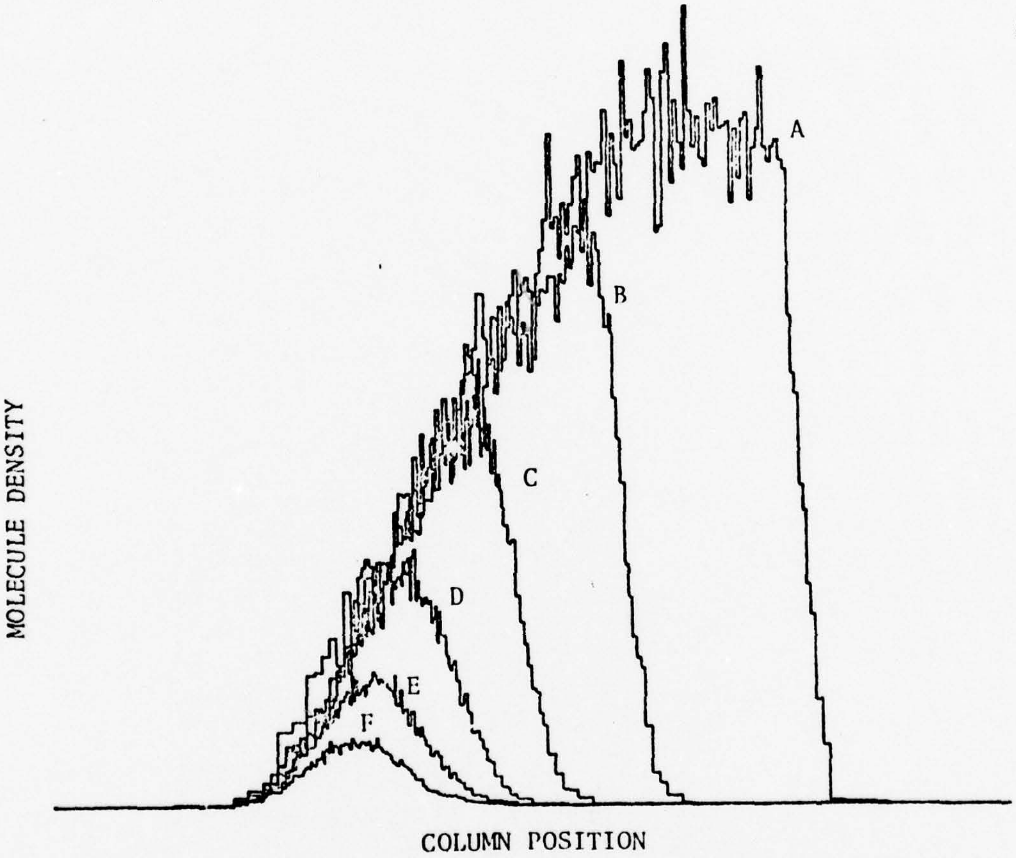


Figure 3.15 Effect of sample size variation on simulation of a nonlinear chromatography model.

Surface encounter rate = .008, desorption rate = .004, plot time = 20000. Sites per molecule = 8, 16, 32, 64, 128, and 256 for plots A, B, C, D, E, and F, respectively.

simulation and has been linear for most of its retention time. But for a short time after injection it had a large enough concentration to move faster than normal and develop a small tail.

The next two peaks are more obviously nonlinear. They have the expected asymmetry with a sharper front and a tail at the rear. The fronts are no longer self-sharpening as they were for a while after injection and the smaller of the two is well on its way to the Gaussian shape.

The two largest peaks still have self-sharpening fronts. The distance a nonlinear peak can travel and still maintain a self-sharpening front is determined by the size of the injection.

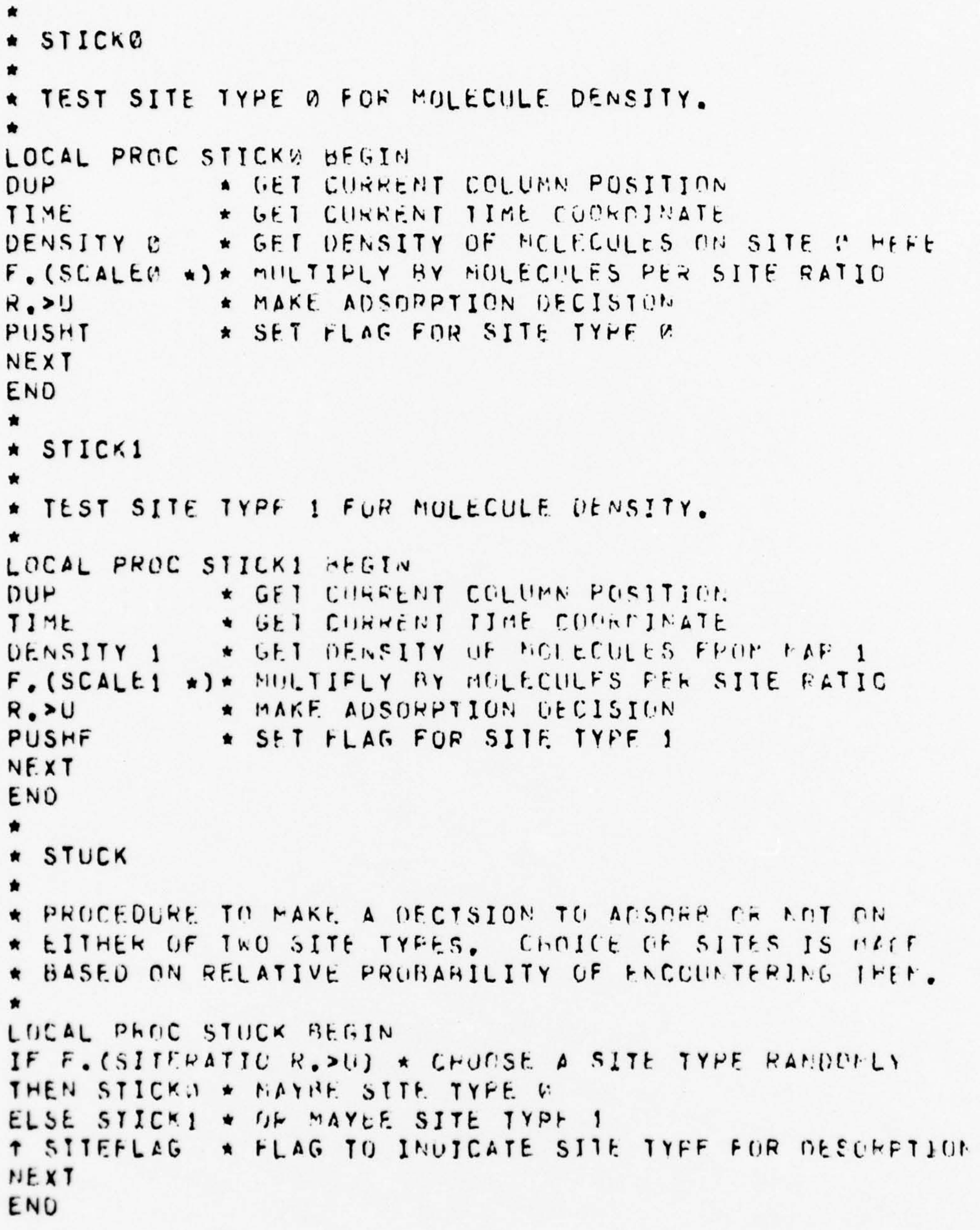
All of the nonlinear peaks in Figure 3.15 follow the same curve on the tail side of the peak. This is a result of the faster average movement of molecules in regions of higher concentration. A molecule in the tail of a peak moves at a rate determined by the number of available adsorption sites which is determined by the local concentration of molecules. Molecules further up the tail are in regions with higher concentration and are, therefore, moving faster and must be on the average pulling away from the rest of the tail. Since molecules in the front are moving away from those in the back, they cannot have any influence on the behavior of those left behind. Thus, molecules in the tail cannot tell how big a peak they belong to and must have the same behavior regardless. This argument assumes that spreading due to diffusion processes is insignificant in comparison with spreading due to nonlinearity.

Nonuniform Surface Models

So far all models simulated have assumed that the solid surface consists of identical discrete adsorption sites. This uniformity of surface structure is an often-stated goal in the design of chromatographic systems (105,106), but real surfaces are always more complicated. For example, a modified Porasil surface intended for gas-solid chromatography was prepared and characterized by Gilpin and Burke (107). It was shown to have three distinct kinds of structures on the surface which could act as adsorption sites for molecules of various types. Any realistic gas-solid chromatography simulation must be able to include these and other nonuniform surface models.

Two Independent Sites

It is quite easy to modify the nonlinear adsorption model given in Figure 3.15 to include the possibility of the molecule becoming adsorbed on a second kind of site with different characteristics. One such nonuniform surface model is given in Figure 3.16. Upon encountering the surface in the same HITSURF procedure, a decision is made as to which of the two types of sites is present. In this particular model, it is assumed that the molecules encounter the sites in proportion to their occurrence frequency on the surface. This may not always be the case, since adsorption kinetics can be influenced by many factors. Once the site type has been determined through the use of a uniform distribution random number, the molecule attempts to adsorb on it. The two adsorption procedures, STICKO and STICK1, are identical to the original nonlinear STUCK procedure from Figure 3.10 except that they use separate



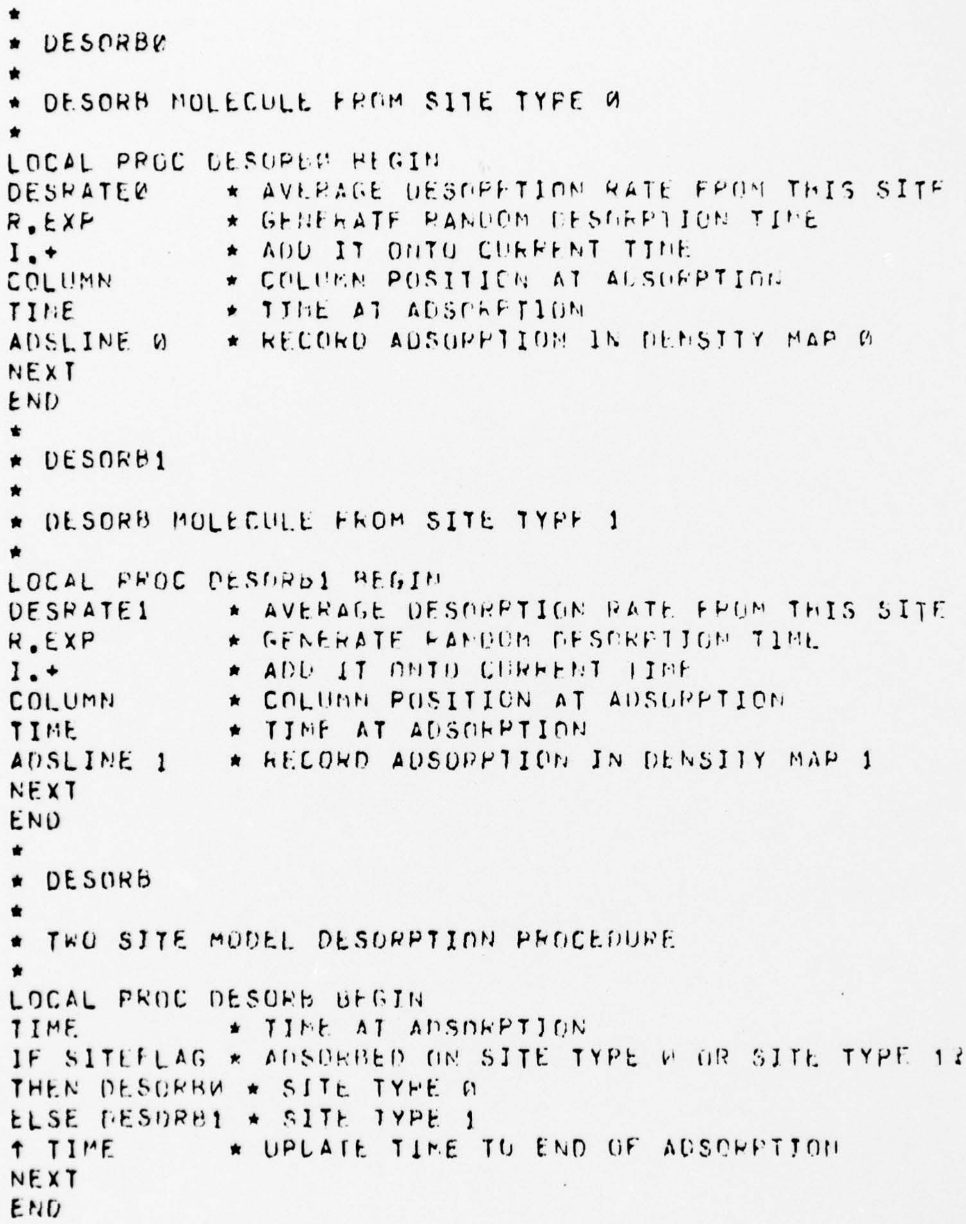
```
*
* STICKØ
*
* TEST SITE TYPE Ø FOR MOLECULE DENSITY.
*
LOCAL PROC STICKØ BEGIN
DUP            * GET CURRENT COLUMN POSITION
TIME           * GET CURRENT TIME COORDINATE
DENSITY Ø      * GET DENSITY OF MOLECULES ON SITE Ø HERE
F.(SCALEØ *)*  MULTIPLY BY MOLECULES PER SITE RATIO
R.>U           * MAKE ADSORPTION DECISION
PUSHT          * SET FLAG FOR SITE TYPE Ø
NEXT
END
*
* STICK1
*
* TEST SITE TYPE 1 FOR MOLECULE DENSITY.
*
LOCAL PROC STICK1 BEGIN
DUP            * GET CURRENT COLUMN POSITION
TIME           * GET CURRENT TIME COORDINATE
DENSITY 1      * GET DENSITY OF MOLECULES FROM MAP 1
F.(SCALE1 *)*  MULTIPLY BY MOLECULES PER SITE RATIO
R.>U           * MAKE ADSORPTION DECISION
PUSHF          * SET FLAG FOR SITE TYPE 1
NEXT
END
*
* STUCK
*
* PROCEDURE TO MAKE A DECISION TO ADSORB OR NOT ON
* EITHER OF TWO SITE TYPES.  CHOICE OF SITES IS MADE
* BASED ON RELATIVE PROBABILITY OF ENCOUNTERING THEM.
*
LOCAL PROC STUCK BEGIN
IF F.(SITERATIO R.>U) * CHOOSE A SITE TYPE RANDOMLY
THEN STICKØ * MAYBE SITE TYPE Ø
ELSE STICK1 * OR MAYBE SITE TYPE 1
↑ SITEFLAG  * FLAG TO INDICATE SITE TYPE FOR DESORPTION
NEXT
END
```

Figure 3.16 Nonlinear adsorption model for surfaces with two kinds of adsorption sites.

density maps and scale factors to make an adsorb or not decision. In a particular region of the molecule density maps, one of the two kinds of sites may be nearly full while the other is almost empty. Whether a molecule becomes adsorbed or not would then depend largely upon which type of site it encountered on the surface. The DESORB procedure for this two-site model is given in Figure 3.17. It generates an appropriate desorption time depending upon which of the two sites the molecule was adsorbed on.

Adding a second type of adsorption site greatly increases the range of possible chromatographic behavior of a model surface. The two individual kinds of sites can have different concentrations and adsorption-desorption kinetics. A molecule in the column can be involved in a larger variety of events and may interact with other molecules in a peak in more complex ways. The explanations for the chromatographic behavior of a system on a molecular level can become much more obscure than the single-site type examples have been.

For example, Figure 3.18 illustrates the effect of varying the proportions of two distinct kinds of adsorption sites on the retention and shape of a chromatographic peak. The adsorption and desorption models used for these simulations are given in Figures 3.16 and 3.17. All plots were made at the same dimensionless time and on the same density scale. The only variation among these simulations was in the amount of higher energy sites present in the column. A higher energy site is one which has a higher activation energy for desorption and, therefore, a slower desorption rate. In this particular example, the



```
*
* DESORBØ
*
* DESORB MOLECULE FROM SITE TYPE Ø
*
LOCAL PROC DESORBØ BEGIN
DESRATEØ      * AVERAGE DESORPTION RATE FROM THIS SITE
R.EXP         * GENERATE RANDOM DESORPTION TIME
I.+           * ADD IT ONTO CURRENT TIME
COLUMN        * COLUMN POSITION AT ADSORPTION
TIME          * TIME AT ADSORPTION
ADSLINE Ø     * RECORD ADSORPTION IN DENSITY MAP Ø
NEXT
END
*
* DESORB1
*
* DESORB MOLECULE FROM SITE TYPE 1
*
LOCAL PROC DESORB1 BEGIN
DESRATE1      * AVERAGE DESORPTION RATE FROM THIS SITE
R.EXP         * GENERATE RANDOM DESORPTION TIME
I.+           * ADD IT ONTO CURRENT TIME
COLUMN        * COLUMN POSITION AT ADSORPTION
TIME          * TIME AT ADSORPTION
ADSLINE 1     * RECORD ADSORPTION IN DENSITY MAP 1
NEXT
END
*
* DESORB
*
* TWO SITE MODEL DESORPTION PROCEDURE
*
LOCAL PROC DESORB BEGIN
TIME          * TIME AT ADSORPTION
IF SITEFLAG * ADSORBED ON SITE TYPE Ø OR SITE TYPE 1?
THEN DESORBØ * SITE TYPE Ø
ELSE DESORB1 * SITE TYPE 1
↑ TIME        * UPDATE TIME TO END OF ADSORPTION
NEXT
END
```

Figure 3.17 Desorption model for surfaces with two kinds of sites.

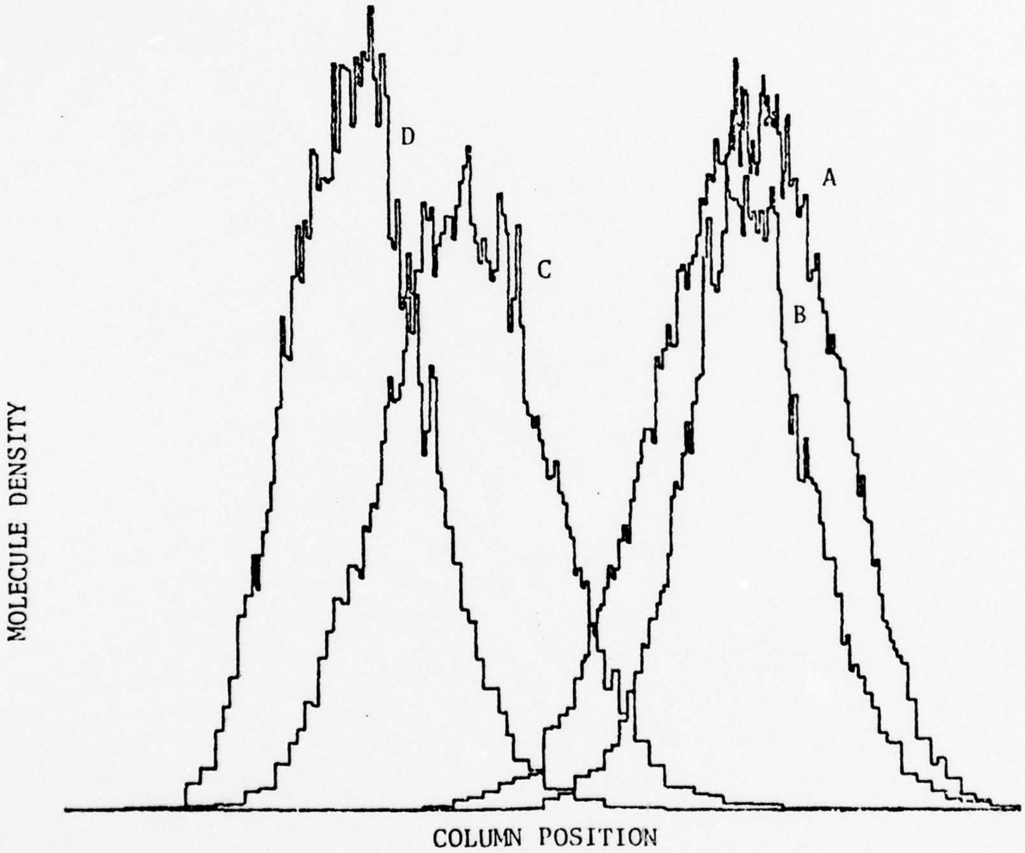


Figure 3.18 Simulation of a nonlinear chromatography model with two types of adsorption sites.

Surface encounter rate = .01, sites 0 desorption rate = .01, sites 1 desorption rate = .001, sites 0 per molecule = 1000, plot time = 25000. Sites 1 per molecule = 0, 10, 100, and 200 for plots A, B, C, and D, respectively.

higher energy site desorption rate was slower than the low energy site by a factor of 10. In all of these simulations the low energy site was present in sufficient concentration to have essentially linear behavior.

The two kinds of sites are assumed to exist independently on the surface and molecules can interact with only one of them at a time. The adsorption rates for each of the two kinds of sites is determined by the surface encounter rate and by the fraction of each kind of site. The probability that a molecule attempts to become adsorbed on one kind of site rather than the other is determined solely by their relative availability and not by any chemical differences.

Plot A of Figure 3.18 resulted from a simulation with no high energy sites. It provides a uniform surface standard against which the two-site models can be compared. Plot B resulted from an identical simulation except for high energy sites added at a concentration 1% as large as the low energy sites. The peak is retained slightly longer in the column and made slightly broader. The effect is not very large simply because at only 1% concentration the molecules do not run into a high energy site very often. With the high energy site concentration set at 10% of the low energy site concentration plot C results. Here the peak has about the same breadth as in plot B, but it has moved only a little more than half as far down the column. The presence of the high energy sites is causing excessive peak broadening and poor column efficiency. At 20% high energy site concentration, plot D, the peak retention has increased by the peak is also sharper than in plot C. The column efficiency has not deteriorated further and may be even a little better as the chromatographic behavior of the column begins to be dominated by the high energy sites.

This simulation experiment confirms the rule that uniform surfaces are better than nonuniform surfaces for good, efficient gas-solid chromatography. It also shows why it is so difficult to make really good gas-solid chromatography surfaces. It is desirable to have low energy nonspecific adsorption as the major retention mechanism in a gas-solid chromatography column. This is often achieved by deactivating the surface through a chemical reaction to remove or cover up the specific interaction high energy adsorption sites. But, if the reaction is not 100% complete and a few high energy adsorption sites, say 5-10%, remain uncovered, then a situation like plot C of Figure 3.18 will result and the column efficiency will be much worse than expected. The results could be considerably worse than this if the high energy adsorption sites have slower desorption rates than were assumed in these models.

The modified Porasil surfaces are of this kind (107). Most of the original silanol groups present on the silica surface were replaced with chemically bonded hydrocarbons which should be very good for low energy nonspecific adsorption processes. But if the original silica surface contained at least 5 silanol groups per nm^2 (108) and the first step of the reaction results in about 3.52 dimethylchlorosilane groups attached per nm^2 (107), then about 1.5 silanol groups per nm^2 must remain unreacted. Many of these are probably effectively covered up by neighboring hydrocarbon groups but the ones remaining are potential specific interaction sites. The concentration of these high energy sites is very likely 5% or more of the total available adsorption sites. If a sample containing functional groups which can bind to these silanol sites is injected, then the resulting chromatographic behavior is likely to be at least as bad as the simulated results.

How tightly a molecule is held on a specific adsorption site depends on the molecule as well as the chemistry of the site. An alkene sample injected into a chromatographic column of this modified Porasil surface should have a moderately strong specific as well as a weak non-specific adsorption. An alkene sample, on the other hand, should not have any specific interactions with the silanol adsorption sites and so should behave as if it is adsorbing on a single-site type of surface. A series of column efficiency measurements was made on modified Porasil surfaces using both cyclohexene and cyclohexane (109). The column efficiency measured using cyclohexene was much worse than the column efficiency measured using cyclohexane. This difference in efficiencies is very neatly explained by the model simulated in Figure 3.18. Cyclohexene should have a moderately strong specific interaction with silanol groups on the silica surface which cyclohexane should not have. Thus, an injection of cyclohexene should see a surface with two kinds of adsorption sites as in plots B, C, and D while an injection of cyclohexane should see only nonspecific adsorption sites as in plot A. A silanol site concentration of 5% to 10% and a cyclohexene desorption rate from these silanol sites 10 to 20 times slower than from nonspecific hydrocarbon sites is consistent with the results of these experiments.

The simulation cannot prove that this model is the correct one. Others may give results just as close to the real experiment. But it does prove that the model is at least consistent with the experimental results. An indication of which models are consistent with experiments can be very valuable in the planning of further research. This is especially true for research involving complex systems such as chromatography where intuition is not enough.

Two Sites with Intersite Transfers

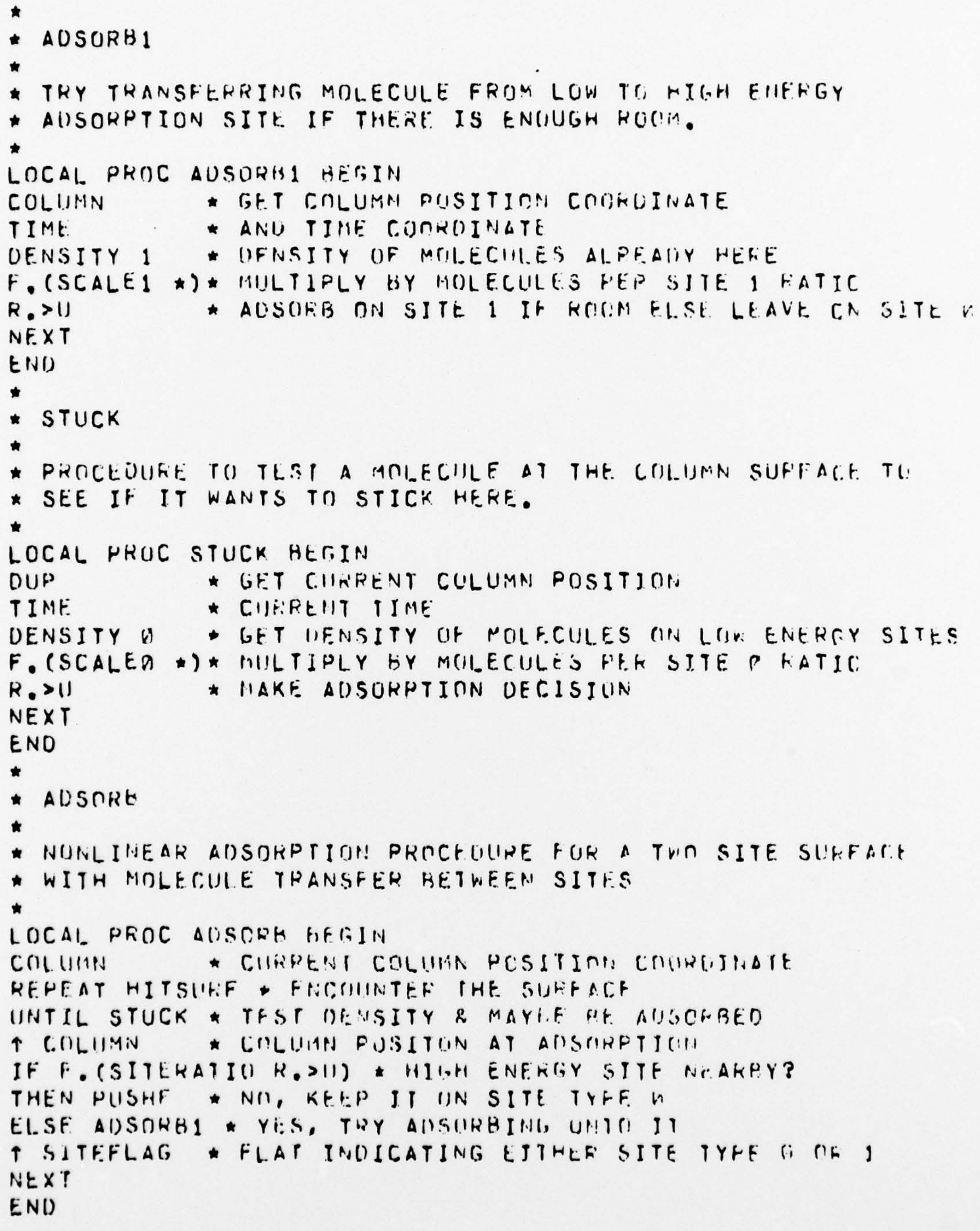
The two-sites model with independent sites is probably good enough if the two kinds of sites are not too different in chemical behavior or concentration. But it may not be adequate if the two sites have fundamentally different character, resulting in different mechanisms of adsorption and desorption. For example, one of the two sites may have a low energy nonspecific adsorption while the second can become involved in a specific interaction with a functional group on an adsorbed molecule. To become adsorbed, the molecule must encounter the surface with its functional group on the side toward the adsorption site. This must happen less often than simply hitting the surface as in a nonspecific adsorption event. Therefore, the kinetics of adsorption on a specific interaction site should be slower than the simple site encounter rate.

However, if there is a large concentration of nonspecific adsorption sites surrounding each specific site, then the molecule may be held on a neighboring nonspecific site while it flops around and eventually falls into the specific site with its functional group in the proper position for adsorption. The activation energy for transfers between adsorption sites is likely to be smaller than for complete removal from the surface in a desorption event and, consequently, the rate of transfer between sites is likely to be faster than the rate of desorption. The adsorbed molecules may then act as a two-dimensional gas on the solid surface (110). While moving over the surface, a molecule could encounter a high energy adsorption site and become stuck on it. Either of these mechanisms involving first a nonspecific adsorption followed by transfer to a specific interaction adsorption site should lead to improved kinetics for adsorption on high energy sites.

A two-site adsorption model involving transfer to high energy sites from neighboring low energy sites is given in Figure 3.19. In this model, it is assumed that a molecule must first become adsorbed on a nonspecific site in the usual fashion. It may then transfer to a neighboring specific interaction site if one is available and empty. It is assumed that on the average each specific interaction site has four nonspecific sites close enough to supply molecules. No possibility of transfers between nonspecific adsorption sites is included in the model.

The result of a simulation of this model is given in Figure 3.20. In this simulation, there is only one high energy site for every 1,000 low energy sites, but once a molecule is adsorbed on a high energy site, it stays there on an average 500 times as long as on a low energy site. The main part of the peak is about where it should be for approximately linear chromatography on the low energy adsorption sites;

70



```
*
* ADSORB1
*
* TRY TRANSFERRING MOLECULE FROM LOW TO HIGH ENERGY
* ADSORPTION SITE IF THERE IS ENOUGH ROOM.
*
LOCAL PROC ADSORB1 BEGIN
COLUMN         * GET COLUMN POSITION COORDINATE
TIME           * AND TIME COORDINATE
DENSITY 1      * DENSITY OF MOLECULES ALREADY HERE
F.(SCALE1 *)* MULTIPLY BY MOLECULES PER SITE 1 RATIO
R.>U           * ADSORB ON SITE 1 IF ROOM ELSE LEAVE ON SITE Ø
NEXT
END
*
* STUCK
*
* PROCEDURE TO TEST A MOLECULE AT THE COLUMN SURFACE TO
* SEE IF IT WANTS TO STICK HERE.
*
LOCAL PROC STUCK BEGIN
DUP            * GET CURRENT COLUMN POSITION
TIME           * CURRENT TIME
DENSITY Ø      * GET DENSITY OF MOLECULES ON LOW ENERGY SITES
F.(SCALEØ *)* MULTIPLY BY MOLECULES PER SITE Ø RATIO
R.>U           * MAKE ADSORPTION DECISION
NEXT
END
*
* ADSORB
*
* NONLINEAR ADSORPTION PROCEDURE FOR A TWO SITE SURFACE
* WITH MOLECULE TRANSFER BETWEEN SITES
*
LOCAL PROC ADSORB BEGIN
COLUMN         * CURRENT COLUMN POSITION COORDINATE
REPEAT HITSURF * ENCOUNTER THE SURFACE
UNTIL STUCK * TEST DENSITY & MAYBE BE ADSORBED
↑ COLUMN       * COLUMN POSITON AT ADSORPTION
IF F.(SITERATIO R.>U) * HIGH ENERGY SITE NEARBY?
THEN PUSHF   * NO, KEEP IT ON SITE TYPE Ø
ELSE ADSORB1 * YES, TRY ADSORBING UNTO IT
↑ SITEFLAG   * FLAT INDICATING EITHER SITE TYPE Ø OR 1
NEXT
END
```

Figure 3.19 Nonlinear adsorption model for surfaces with two kinds of sites and surface transfer of molecules from low energy to high energy sites.

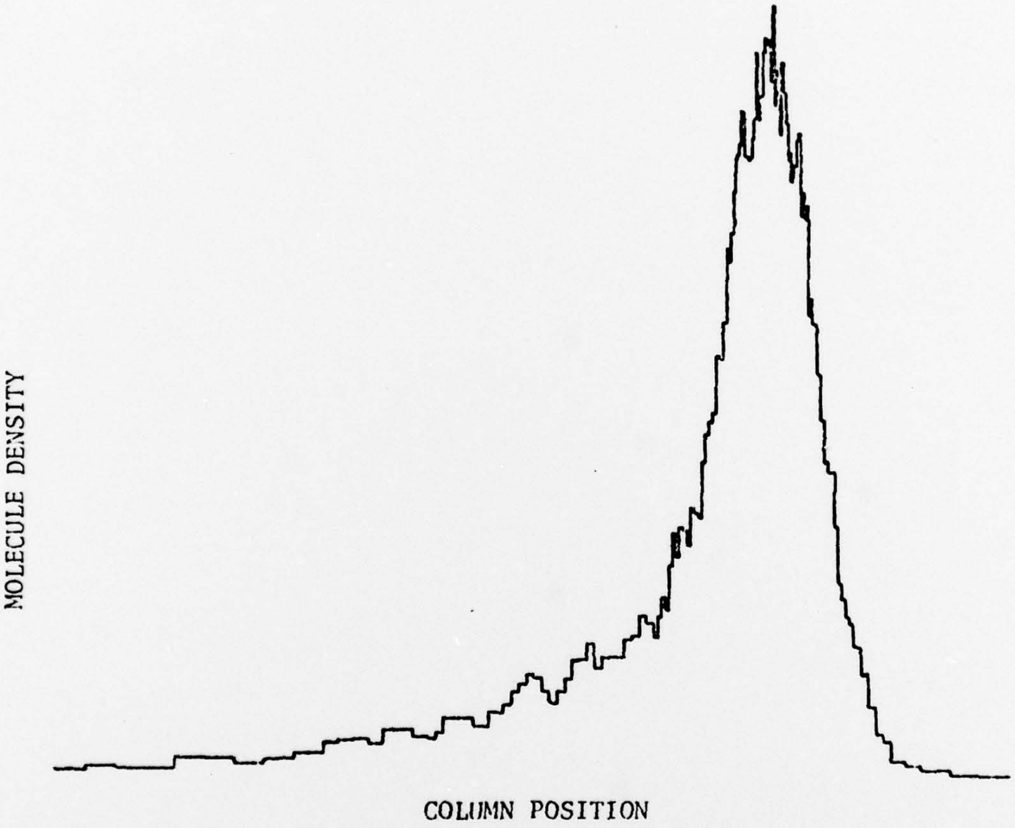


Figure 3.20 Simulation of a nonlinear chromatography model with two types of adsorption sites and surface transfer of molecules from low energy to high energy sites.

Surface encounter rate = .02, sites 0 desorption rate = .02, sites 1 desorption rate = .001, sites 0 per molecule = 400, sites 1 per molecule = 4, plot time = 25000.

therefore, most of the molecules must be completely ignoring the high energy sites.

The peak tail has a very different shape than the tails resulting from simply overloading a single-site type column. It is much longer because of the very slow desorption kinetics of the high energy sites and lower in conccentration because of the limited number of these sites. The adsorption kinetics are enhanced by the mechanism transferring molecules from low to high energy sites. The high energy sites must be nearly saturated wherever there is a reasonably large concentration of molecules adsorbed on the low energy sites. As the main part of the peak passes over a section of the column, almost all of the high energy sites are filled. Then, after the peak has moved on, molecules slowly desorb from the high energy sites. But most of them will never catch up with the main peak because it is moving at a rate mainly determined by the low energy sites while out in the tail the high energy sites are no longer completely saturated. Consequently, a molecule has an increased chance of becoming adsorbed on a high energy site again and moving even further out in the tail.

This very low concentration of high energy sites continuously builds a tail on the chromatographic peak at a rate determined by the concentration of the sites. The length of the tail is determined by the desorption kinetics. To be effective tail producers, high energy sites must have some adsorption mechanism which increases the rate over what it would be if they became adsorbed by simply encountering the site directly from the gas phase.

Giddings (13) argued that in order to produce a distinct tail on a peak a high energy site must have a desorption rate at least 10^5 times slower than the low energy sites responsible for the main part of the peak. This is clearly not true for this model since the high energy site desorption rate is only 500 times the low energy site desorption rate. Giddings did not consider the possibility of a molecule being moved further out into the tail by additional adsorptions on the high energy sites. The enhanced rate of adsorption in this model makes these additional adsorptions in the tail an important part of the mechanism responsible for building the tail to a significant size. In a two-dimensional gas model, the rate of adsorption on the high energy sites might be even greater, resulting in a faster buildup of a tail.

This kind of model can help explain the lack of peak tailing in cross-correlation chromatography experiments discussed in Chapter 2. In these multiple injection experiments, each peak is injected on the tail of, or even overlapping with, the previously injected peak. Each sample injected finds a surface with all of its higher energy sites full from the previous injection and so cannot use this mechanism to any significant degree in building a tail.

Figure 3.21 illustrates the effect of changing the sample size on the model given in Figure 3.18. Plot B is the same simulation presented in Figure 20. Plot A is an increase in sample size by a factor of 4 and plot C is a decrease by a factor of 4. The attenuations of plots A and C are adjusted to the same scale as plot B. All three simulations are plotted at the same simulation time.

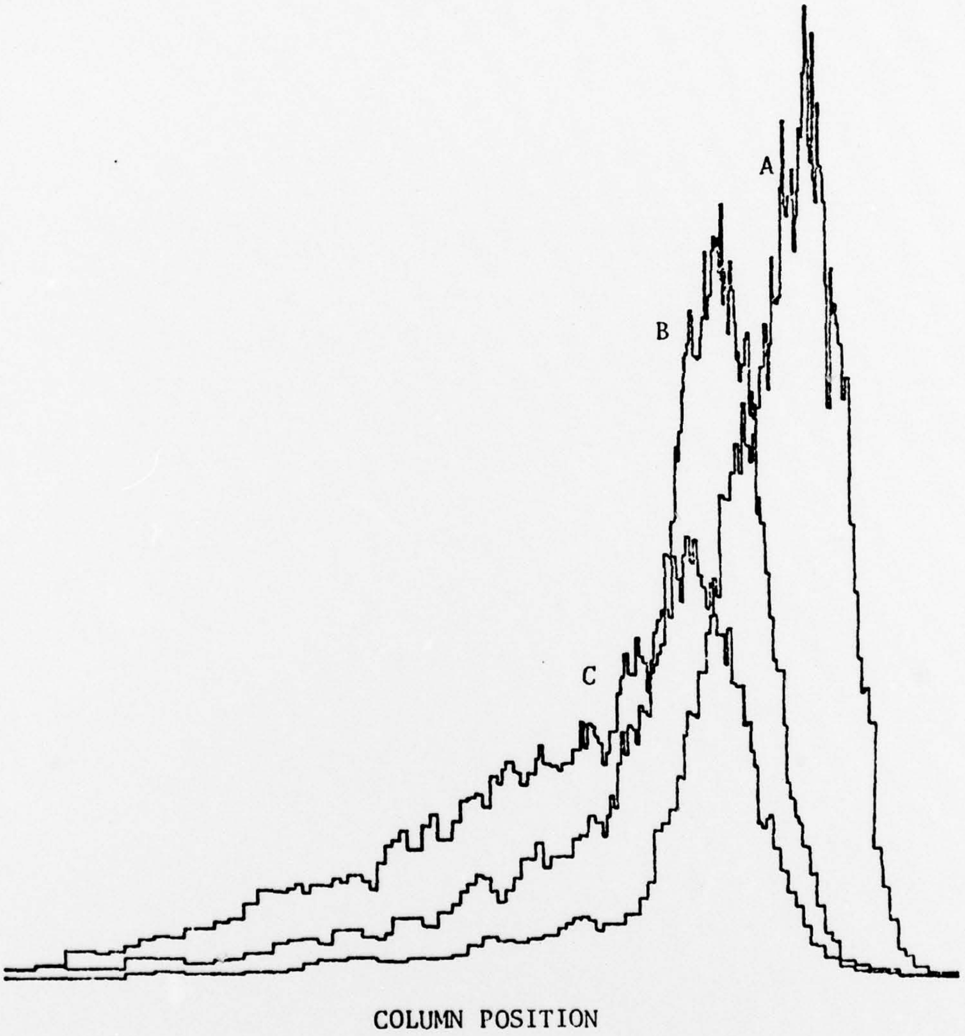


Figure 3.21 Effect of sample size variation on simulation of a non-linear chromatography model with two types of adsorption sites and surface transfer of molecules from low energy to high energy sites.

Surface encounter rate = .02, sites 0 desorption rate = .02, sites 1 desorption rate = .001, plot time = 25000. Sites 0 per molecule = 100, 400, and 1600 and sites 1 per molecule = 1, 4, and 16, for plots A, B, and C, respectively.

Increasing the sample size four times causes the low energy adsorption sites to begin behaving nonlinearly. The result is a shift to shorter retention times and a larger column overloading tail. As expected, increasing the sample size is detrimental to the efficiency of this gas-solid chromatography column. But decreasing the sample size is also bad. Now the high energy tail producing sites can take a larger proportion of the molecules from the main part of the peak and move them into the tail. This kind of tail is limited by the number of high energy sites and as the total sample size is reduced, a small number of sites becomes proportionally more important. Giddings (111) has previously observed and explained this kind of behavior.

Conclusion

This simulation system is not an attempt to create a new theory of chromatography. From the point of view of a pure theoretician, a simulation is an inelegant brute force procedure woefully lacking in generality. A specific model must be chosen for each simulation experiment and the results can be generalized to a class of models only through interpretation of the experimental (simulated) evidence. From this point of view, a simulation procedure more closely resembles an experimental technique than a theory. But, to an experimentalist, a simulation may seem to lack any contact with the real world and its experimentally verifiable facts. Actually, simulation methods lie somewhere between theory and experiment and can be usefully employed in combination with either or both. They can help link theory and experiment together to improve the understanding of both, indicating which elements of a theory can be used to explain an experimental result or what kinds of experiments should be employed to verify a theory.

Any model or theory which can be expressed as in terms of the discrete event algorithms and does not exceed the capacity of the available computer may be simulated. Many models which are difficult to express and solve analytically may be examined by this simulation procedure. The basic requirement is that the model be expressable in terms of mechanisms and probabilities. For example, diffusion of a molecule leaving a surface and returning to the gas phase and flowing through porous structures may be modelled by probability density functions derived from the geometry of gas flow through the porous structure. This case has been treated by Giddings (112) only by making very restrictive assumptions. Such assumptions are not required by this simulation approach.

This simulation procedure operates on kinetic models. Thermodynamic models may be simulated only if they can be reformulated in kinetic terms. Thermodynamics is concerned with the initial and final states of a system while kinetics is concerned with the details of how a system gets from one state to another. In principle, all thermodynamic models may be treated in terms of kinetics although the kinetic models may be more complex and less accurate. For example, a thermodynamic partition coefficient may be defined as the equilibrium ratio of the stationary phase activity to the mobile phase activity of the sample substance. In a kinetic model, this partition coefficient may be defined as a ratio of the adsorption and desorption rates and ideally the two models should agree. If one desires to model the temperature dependence, a thermodynamic parameter, of a chromatographic system, then some model of the temperature dependence of the kinetic parameters must be provided. Desorption rates are easy to model using the Arrhenius equation. Surface encounter rates, however, are composites of several processes, each of which may have a different temperature dependence. But, if the user can derive satisfactory submodels and is willing to make his chromatography model more complex, he can simulate partition coefficient temperature dependences or any other thermodynamic processes.

No attempt has been made to derive quantitative results from any of these simulations of chromatographic processes. It would be possible to assign real values to the dimensionless parameters used in the models and so scale the results to match some real chromatographic system. To do so might be helpful to the reader in developing more realistic mental pictures of the chromatographic systems being modelled. The results, however, would not be immediately useful for quantitative parameter estimation. The models as defined are still too simple to completely describe real chromatographic systems. Adding more detail necessarily adds more parameters whose values must be estimated. With enough parameters to adjust, almost any experimental result could be matched with a wide variety of models.

The models presented here are realistic in terms of their chromatographic behavior. However, the submodels of the structure of gas-solid surfaces and adsorption-desorption processes in chromatographic systems are grossly oversimplified. Many details of molecules interacting with surfaces have been left out of these models to make computer simulation of them practical. Generally, large numbers of individual discrete events have been replaced by single random numbers from distributions which are assumed to have the same overall result. The simple adsorption events of these models are in reality complex composites of flow between particles, diffusion through pores, actual adsorption, and surface diffusion processes. A real numerical view for such a composite parameter is little more enlightening than the dimensionless parameters.

The greatest value of these digital computer simulation techniques is in their use as qualitative aids to the understanding of models of chromatographic mechanisms. Informal models are a very important part of the thinking processes of all chemists engaged in research on new chemical systems. Very often, as in the case of chromatography, these informal models tend to be based on molecular interactions mechanisms. A discrete event simulation is a more formal way of expressing the same kind of molecular model. In exchange for the extra effort required in formally defining and writing down a model, the simulation provides a way of testing the logic of a model. The testing can provide evidence as to the reasonableness of a model and so aid in the intelligent discussion and comparison of alternative models leading to valuable insight into the behavior of corresponding real chromatographic systems.

TECHNICAL REPORT DISTRIBUTION LIST

Dr. M. B. Denton
Department of Chemistry
1 University of Arizona
Tucson, AZ 85721

Dr. G. S. Wilson
Department of Chemistry
1 University of Arizona
Tucson, AZ 85721

Dr. R. A. Osteryoung
Department of Chemistry
1 Colorado State University
Fort Collins, CO 80521

Dr. B. R. Kowalski
Department of Chemistry
1 University of Washington
Seattle, WA 98105

Dr. I. B. Goldberg
North American Rockwell Science Center
1 P. O. Box 1085
1049 Camino Dos Rios
Thousand Oaks, CA 91360

Dr. S. P. Perone
Department of Chemistry
1 Purdue University
Lafayette, IN 47907

Dr. E. E. Wells
Naval Research Laboratory
1 Code 6160
Washington, DC 20375

Dr. D. L. Venezky
Naval Research Laboratory
1 Code 6130
Washington, DC 20375

Dr. Fred Saalfeld
Naval Research Laboratory
1 Code 6110
Washington, DC 20375

Dr. H. Chernoff
Massachusetts Institute of Technology
1 Department of Mathematics
Cambridge, MA 02139

Dr. K. Wilson
Department of Chemistry
1 University of California, San Diego
La Jolla, CA 92037

Dr. A. Zirino
1 Naval Undersea Center
San Diego, CA 92132

Dr. John Duffin
1 United States Naval Post Graduate School
Monterey, CA 93940

Dr. G. M. Hieftje
Department of Chemistry
1 Indiana University
Bloomington, IN 47401

Dr. Victor L. Rehn
Naval Weapons Center
1 Code
China Lake, CA 93555

Office of Naval Research
2 Arlington, VA 22217
Attn: Code 472

Office of Naval Research
6 Arlington, VA 22217
Attn: Code 1021P

ONR Branch Office
536 South Clark Street
1 Chicago, IL 60605
Attn: Dr. George Sandoz

ONR Branch Office
715 Broadway
1 New York, NY 10003
Attn: Scientific Department

ONR Branch Office
1030 East Green Street
1 Pasadena, CA 91106
Attn: Dr. R. J. Marcus

ONR Branch Office
1 760 Market Street, Room 447
San Francisco, CA 94102
Attn: Dr. P. A. Miller

ONR Branch Office
1 495 Summer Street
Boston, MA 02210
Attn: Dr. L. H. Peebles

Director, Naval Research Laboratory
Washington, DC 20390
6 Attn: Library, Code 2029 (ONRL)
1 Technical Information Division
1 Code 6100, 6170

The Assistant Secretary of the Navy (R&D)
1 Department of the Navy
Room 4E736, Pentagon
Washington, DC 20350

Commander, Naval Air Systems Command
1 Department of the Navy
Washington, DC 20360
Attn: Code 310C (H. Rosenwasser)

Defense Documentation Center
2 Building 5, Cameron Station
Alexandria, VA 22314

U. S. Army Research Office
1 P. O. Box 12211
Research Triangle Park, NC 27709
Attn: CRD-AA-IP

Commander
1 Naval Undersea Research & Development Center
San Diego, CA 92132
Attn: Technical Library, Code 133

Naval Weapons Center
1 China Lake, CA 93555
Attn: Head, Chemistry Division

Naval Civil Engineering Laboratory
1 Port Hueneme, CA 93041
Attn: Mr. W. S. Haynes

Professor O. Heinz
1 Department of Physics & Chemistry
Naval Postgraduate School
Monterey, CA 93940

Dr. A. L. Slafkosky
1 Scientific Advisor
Commandant of the Marine Corps (Code RD-1)
Washington, DC 20380